



**This electronic thesis or dissertation has been  
downloaded from Explore Bristol Research,  
<http://research-information.bristol.ac.uk>**

*Author:*

**Miller, John Francis**

*Title:*

**The determination of very small electrophoretic mobilities of dispersions in non-polar media using phase analysis light scattering.**

#### **General rights**

The copyright of this thesis rests with the author, unless otherwise identified in the body of the thesis, and no quotation from it or information derived from it may be published without proper acknowledgement. It is permitted to use and duplicate this work only for personal and non-commercial research, study or criticism/review. You must obtain prior written consent from the author for any other use. It is not permitted to supply the whole or part of this thesis to any other person or to post the same on any website or other online location without the prior written consent of the author.

#### **Take down policy**

Some pages of this thesis may have been removed for copyright restrictions prior to it having been deposited in Explore Bristol Research. However, if you have discovered material within the thesis that you believe is unlawful e.g. breaches copyright, (either yours or that of a third party) or any other law, including but not limited to those relating to patent, trademark, confidentiality, data protection, obscenity, defamation, libel, then please contact: [open-access@bristol.ac.uk](mailto:open-access@bristol.ac.uk) and include the following information in your message:

- Your contact details
- Bibliographic details for the item, including a URL
- An outline of the nature of the complaint

On receipt of your message the Open Access team will immediately investigate your claim, make an initial judgement of the validity of the claim, and withdraw the item in question from public view.

THE DETERMINATION OF VERY SMALL  
ELECTROPHORETIC MOBILITIES OF DISPERSIONS  
IN NON-POLAR MEDIA USING PHASE ANALYSIS  
LIGHT SCATTERING

by

JOHN FRANCIS MILLER

A thesis submitted to the University of Bristol in accordance  
with the requirements for the degree of Doctor of Philosophy  
in the Faculty of Science.

Department of Physical Chemistry, University of Bristol,  
November 1990.

**BEST COPY**

**AVAILABLE**

**PAGE  
NUMBERS  
CUT OFF  
IN  
ORIGINAL**



### ABSTRACT

An apparatus is described that can determine electrophoretic mobilities of polar and non-polar colloidal dispersions down to  $\sim 10^{-12} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$ , with typical resolutions of 0.5 to 5%, depending on the nature of the dispersion being studied. The diffusion coefficient and settling/convection velocity of the sample may be determined simultaneously in real time with the electrophoretic mobility. The technique, phase analysis light scattering (PALS), is based upon classical laser-Doppler electrophoresis, but employs signal processing of the time domain phase information within the scattered light signal, rather than analysis of its frequency spectrum. This allows much smaller electric field strengths to be employed, thereby alleviating the usual heating problems associated with electrophoretic studies of non-polar dispersions. PALS measurements of typical aqueous latex dispersions with large mobilities and non-polar dispersions with very small mobilities (down to  $5 \times 10^{-12} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$ ) are presented to illustrate the versatility of the technique.

### ACKNOWLEDGEMENTS

I would like to thank my supervisor, Dr B Vincent, for his continued enthusiasm and support throughout the period of this project. I am indebted to Professor K Schätzel for the many stimulating discussions, notably concerning the more difficult aspects of the PALS theory. Drs T Cosgrove and S Lubetkin are thanked for their advice and generous loan of equipment during the early stages of the project.

The staff of the mechanical and electronic workshops within the School of Chemistry are thanked for their practical assistance, as is Steve Neck for his apparently neverending ability to keep vital pieces of lab equipment operational, especially the PCS laser.

Drs K Johnson (Glaxo) and I MacDonald (Exxon) are thanked for supplying many of the samples studied during this work (the drugs and the engine oil additives, respectively).

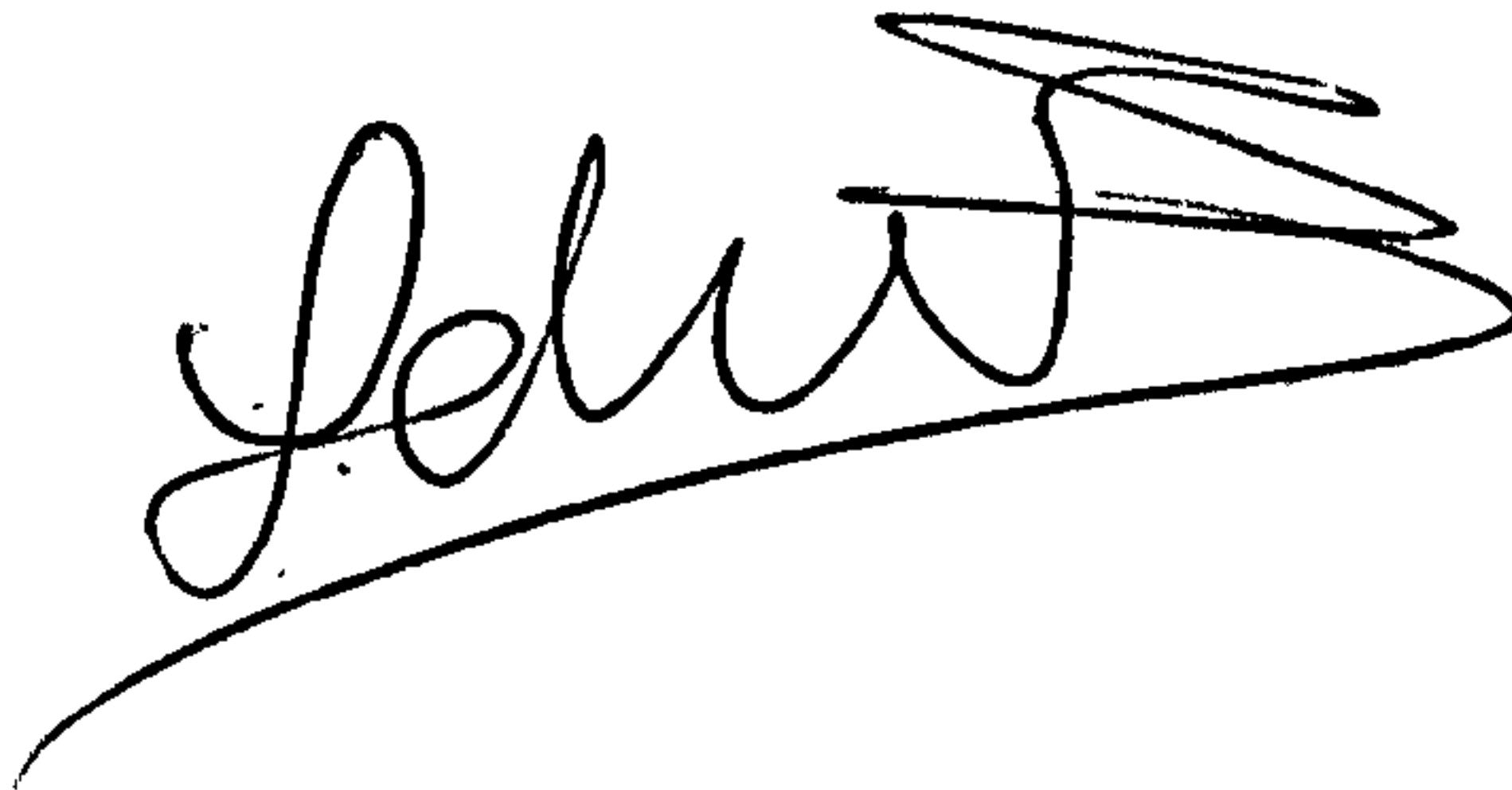
I am extremely grateful to the SERC and Glaxo Group Research for providing me with the means to undertake this project. In particular, Dr D Wyatt (Glaxo) is thanked for his support and continued faith in the project, which, at times, must have appeared to be an expensive gamble.

Thanks go to all my friends and colleagues in Bristol who have made my stay here so enjoyable and worthwhile.

Finally, I must thank Lynne Neck for performing such a wonderful job of typing the manuscript, especially the equations!

# DECLARATION

The work described in this thesis was carried out in the Department of Physical Chemistry, University of Bristol during the period October 1987 to September 1990, under the supervision of Dr B Vincent. Except where acknowledged, the work is original and has not been submitted for any other degree.

A handwritten signature in black ink, appearing to be 'John', written in a cursive style with a long horizontal flourish underneath.

## CONTENTS

	<u>PAGE</u>
CHAPTER 1 : INTRODUCTION	1
CHAPTER 2 : COLLOIDAL DISPERSIONS IN NON-AQUEOUS MEDIA	5
2.1 Colloid Stability	5
2.2 Electrostatic stabilisation in polar media	7
2.2.1 The electrical double layer	7
2.2.2 The interaction between two approaching particles' double layers	10
2.2.3 The interpretation of stability from DLVO plots	11
2.3 Non-aqueous colloidal dispersions	13
2.3.1 Application of DLVO theory	13
2.3.2 Origin of surface charge in non-polar dispersions	18
2.4 Estimation of surface potential and charge via electrokinetic phenomena	26
2.4.1 Electrokinetic phenomena	26
2.4.2 Determination of zeta potentials using electrophoresis	28
2.5 Classical methods for the determination of electrophoretic mobilities	37
2.5.1 Extension to non-aqueous dispersions	40
2.5.2 Electrophoresis in alternating electric fields	44

CHAPTER 3 : LIGHT SCATTERING	46
3.1 Basic principles of light scattering	46
3.1.1 Scattering from small particles	46
3.1.2 Scattering from large particles	47
3.2 Scattering from a colloidal dispersion	48
3.3 Photon correlation spectroscopy	50
3.4 Laser-Doppler methods	58
3.4.1 Relating power spectra to correlation functions	58
3.4.2 Obtaining Doppler spectra by heterodyning (light beating)	59
3.4.3 Influence of a constant, collective velocity	61
3.4.4 Application to samples with more than one species of particle	63
3.4.5 Dual beam/real fringe heterodyne velocimeter	64
3.4.6 Fringe model for dual beam differential heterodyne velocimeter	66
3.5 Processing the Doppler signal from a real fringe velocimeter	67
3.5.1 Fringe volume spacial illumination profile	67
3.5.2 Processing signals due to single particles	68
3.5.3 Signals due to many particles	68
3.5.4 Pedestal removal	72
3.5.5 Modulation of the Doppler signal by light frequency shifting	73
3.5.6 Practical limitations of the moving fringe LDE technique	75

3.6	Phase analysis light scattering	77
3.6.1	Relating particle displacements to optical phase changes	78
3.6.2	The two-time distribution of the complex Doppler signal and its importance	80
3.6.3	The amplitude-weighted phase difference (AWPD) function	86
3.6.4	The amplitude-weighted phase structure (AWPS) function	88
3.6.5	AWPD vs. AWPS functions for for electrophoretic mobility determination	92
3.6.6	Minimisation of the constant velocity term	93
3.6.7	Correlation functions for the real/imaginary part of the complex Doppler signal	95
3.6.8	Correlation functions for the amplitude of the complex Doppler signal	97
3.6.9	Probability density functions for the amplitude of the complex Doppler signal	99
3.7	Chapter summary	99
CHAPTER 4	: INSTRUMENT DEVELOPMENT AND VALIDATION	100
4.1	Basic static fringe velocimeter	100
4.1.1	Optical components	100
4.1.2	Photodetector	102
4.1.3	Electrode assembly	103
4.1.4	Sample cell temperature control	104
4.1.5	Signal processing	105
4.1.6	General performance of the static fringe velocimeter	106



4.1.7	Non-aqueous studies with the basic velocimeter	108
4.2	Development of the PALS equipment	109
4.2.1	Overview	109
4.2.2	Generation of moving interference fringes	110
4.2.3	Conventional processing using the moving fringe velocimeter	115
4.2.4	Measurement of amplitude and phase information - the principles	118
4.2.5	Initial phase-sensitive detector hardware used for PALS apparatus	121
4.2.6	Analogue-to-digital converter requirements	122
4.2.7	Selection of programming language and general software overview	124
4.2.8	Preliminary software testing	126
4.2.9	Initial instrument validation	127
4.2.10	Preliminary studies of non-polar dispersions using PALS	130
4.2.11	Incorporation of a lock-in amplifier for phase-sensitive detection	132
4.2.12	Electrode drive signal characterisation	135
4.2.13	Constant velocity term minimisation	136
4.3	Final instrument validation	138
CHAPTER 5	: ELECTROPHORETIC STUDIES OF VARIOUS NON-AQUEOUS DISPERSIONS USING PALS	146
5.1	General methods	146
5.1.1	Particle size determination	146



5.1.2	Water concentration determination	147
5.1.3	Particle mass fraction determination	148
5.1.4	Cleaning of glassware	148
5.1.5	Solvent drying procedure	149
5.1.6	Dust removal	149
5.1.7	Preparation of samples for measurement	150
5.2	Microgel latex preparation	151
5.2.1	General description	151
5.2.2	Preparation of PSLX-PP-1	153
5.2.3	Preparation of PSLX-A-2	154
5.2.4	Redispersal of microgels into non-polar media	154
5.3	Electrophoretic studies of PSLX-PP-1 in toluene	155
5.3.1	Gravimetric analysis of "parent" PSLX-PP-1/toluene dispersions	155
5.3.2	Sample preparation and measurement	155
5.4	Study of PSLX-PP-1 and PSLX-A-2 in dioxan/water mixtures	159
5.5	Study of PSLX-PP-1 in dioxan/water/KCl solutions	162
5.6	Miscellaneous electrophoretic studies of non-polar drug dispersions	164
5.7	Electrophoretic studies of highly viscous non-polar dispersions	167

CHAPTER 6 : DISCUSSION, CONCLUSIONS AND FURTHER WORK	170
6.1 Appraisal of the PALS technique	170
6.2 Suggestions for further work	172
APPENDIX 1 : The statistical accuracy of AWPD and AWPS functions	A1-1
APPENDIX 2 : Software listing and description	A2-1
REFERENCES	R1

## CHAPTER ONE

### INTRODUCTION

Colloid chemistry plays a vital role in modern pharmaceutical drug delivery systems. Targetting of the active material to the required part of the patient's body in a physiologically active form is of paramount importance. For example, "Salbutamol" is a drug widely used for the treatment of asthma. It functions by binding with suitable receptors in the patient's lungs, stimulating bronchial dilation, thereby allowing the patient to breathe more easily. The obvious choice for delivery of the drug is by inhalation. Indeed, other methods, such as the use of oral tablets, may lead to a significant proportion of the drug being delivered to other parts of the body, possibly causing undesirable side effects. The drug is inhaled in the form of particles typically a few microns in size. Whichever the method of delivering the drug particles, it is important that they remain discrete, so as to reach the correct part of the patient's lungs.

In the case of "Ventolin", a trade name for a system developed and marketed by Glaxo Group Research, an aerosol delivery system is employed. The drug is dispersed in a non-polar, highly volatile liquid (a low molecular weight chlorofluorocarbon) and

contained within a pressurised aluminium cannister. The drug is administered by actuation of a suitable metering valve, which causes a controlled volume of the dispersion to spray into the patient's mouth. As this spray travels towards the back of the mouth, the dispersing medium rapidly evaporates, leaving a fine aerosol of drug particles to be inhaled. It is important that the drug dispersion remains stable whilst in the pressurised container in order to prevent valve blockage and to maximise the efficiency of the administered dose.

Surfactants are employed to improve the stability of such dispersions. The surfactants used must be physiologically inactive and chemically inert with respect to the active ingredient. Oleic acid is used in the Ventolin system, and it is possible that adsorption of the surfactant onto the drug promotes charge development at the solid/liquid interface, thereby increasing the kinetic stability of the dispersion (see Chapter 2).

It is known empirically that the presence of trace amounts of water in the dispersion medium can have detrimental effects on the stability of the drug formulation. Thus, at the present time, rigorous drying procedures are employed in order to keep moisture levels in the dispersions below 15 p.p.m. by mass. However, an investigation into the effects of varying quantities of trace water on the kinetic stability and interfacial charge/potential properties of the

dispersions is of interest. Although the determination of kinetic stability of these systems is, in principle, relatively straight forward, the determination of interfacial charge/potential in such systems is difficult, for reasons that will be described later. The study of the effects of water on the drug systems may provide the drug formulators with useful information. However, it will not necessarily increase the colloid chemist's knowledge of the factors controlling the influence of water on colloidal systems in non-polar media. Hence, a study of the effects of water on the stability and surface charge/potential of model systems, such as silica particles or swollen, cross-linked polystyrene microgel particles dispersed in non-polar media, is of interest. Although attempts have been made in the past to investigate this field (Chapter 2), the measurement of the surface charge/potential was difficult and unreliable. The work described in this thesis has primarily been concerned with the development of an apparatus with which one can reliably and accurately determine the electrophoretic velocity of particles dispersed in non-polar media, from which the interfacial charge/potential can be estimated. A model colloidal system, consisting of cross-linked polystyrene microgel particles dispersed in apolar media has been studied with regard to the effects of trace water present in the dispersion on the electrophoretic mobility of the particles. It will be seen that the technique used to measure the electrophoretic mobilities, "phase analysis light scattering" (PALS), can resolve

mobilities as low as  $10^{-12} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$ . The relationship between electrophoretic mobility and surface charge/potential is discussed in the next chapter. Chapter 3 describes the theory of classical dynamic light scattering and phase analysis light scattering. The development and validation of the PALS apparatus is presented in Chapter 4.

The preliminary studies on some non-polar dispersions are described in chapter 5, where it will be seen that the results obtained cannot be faithfully ascribed to the effects of water alone. A discussion of the new technique, its advantages and disadvantages, is presented in chapter 6, along with comments on the preliminary studies of water on non-polar systems.



## CHAPTER TWO

### COLLOIDAL DISPERSIONS IN NON-AQUEOUS MEDIA

#### 2.1 COLLOID STABILITY<sup>1,2</sup>

Most colloidal dispersions are thermodynamically unstable, due to interparticle van der Waals attractive forces. Repulsive forces need to be generated between particles in order to prevent aggregation. The repulsive force is usually invoked either by adsorbing a suitable macromolecular material onto the particle surface, so-called "steric stabilisation"<sup>3,4</sup>, or by the generation of a suitable interfacial charge/potential, so-called "electrostatic stabilisation". Colloid stability is usually interpreted in terms of the change in the interaction energy between two particles as they approach. The estimation of the necessary attractive and repulsive energy contributions to the total interaction energy is the basis of the theory of Deryaguin and Landau<sup>5</sup> and Verwey and Overbeek<sup>6</sup>, referred to as "DLVO theory".

To a first approximation, the attractive force between two particles arises through induced-dipole--induced-dipole interactions<sup>7</sup>. Hamaker<sup>8</sup> has derived the following equation for the attractive energy

term,  $V_A$ , between two spherical colloidal particles as a function of their separation,  $H$ , in vacuo,

$$V_A(H) = \frac{-A}{12} \left[ \frac{y}{x^2 + xy + x} + \frac{y}{x^2 + xy + x + y} + 2 \ln \left( \frac{x^2 + xy + x}{x^2 + xy + x + y} \right) \right] \quad (2.1)$$

where  $x = H/(a_1 + a_2)$  and  $y = a_1/a_2$ , with  $a_1$  and  $a_2$  being the radii of the two particles.  $A$  is the Hamaker constant. A dispersion medium influences the Hamaker constant and, therefore, affects the attraction term. An effective Hamaker constant is calculated from the Hamaker constants for the particle,  $A_{pp}$ , and the medium,  $A_{mm}$ , using

$$A = (A_{pp}^{\frac{1}{2}} - A_{mm}^{\frac{1}{2}})^2 \quad (2.2)$$

The importance of eq. (2.2) is that it shows that the attraction between two particles in a dispersion depends on the similarity of the Hamaker constants. Theoretically, when  $A_{pp} = A_{mm}$ , there will be no attractive force. Particles and media of similar chemical properties tend to have similar Hamaker constants. Exploitation of this phenomenon proves very useful for the model system studied in this work.



## 2.2 ELECTROSTATIC STABILISATION IN POLAR MEDIA

### 2.2.1 THE ELECTRICAL DOUBLE LAYER

When a hydrophobic colloidal particle is immersed in a polar medium, a separation of charge may occur at the interface.

A number of mechanisms may operate, including

- (i) the ionisation of ionogenic groups at the particle surface, e.g.



- (ii) ion adsorption, e.g.



If the charged particle is dispersed in an aqueous electrolyte solution, it is necessary to consider how the potential and the charge are distributed. This is normally described by the classical electrical double layer theory<sup>9, 10, 11</sup>.

The charge distribution at the particle surface is considered to be planar (assuming the particle surface to be planar), whereas the charge in the disperse phase, arising through randomly diffusing

ionic species, will exhibit a Boltzmann-type distribution away from the surface, becoming zero at large distances from the interface. Thus, for a positively charged surface, negative counterions will tend to migrate towards the interface, causing a higher concentration of anions near the particle than away from it. The converse will be true for positively charged "co-ions".

If  $\psi_0$  is the electrical potential at the particle surface and  $r$  is the distance from the surface, the potential at a distance  $r$ ,  $\psi(r)$ , is assumed to follow a Poisson type relationship,

$$\nabla^2 \psi(r) = \Delta \psi(r) = \frac{-\rho(r)}{\epsilon_0 \epsilon_r} \quad (2.3)$$

where  $\nabla = \left( \frac{\delta}{\delta r}, \frac{\delta}{\delta \theta}, \frac{\delta}{\delta \phi} \right)$ ,  $\delta \theta = \delta \phi = 0$ ,  $\epsilon_0$  is the permittivity of free space,  $\epsilon_r$  is the relative permittivity of the medium and  $\rho(r)$  is the charge density at  $r$ .  $\rho(r)$  is determined by the Boltzmann-type distribution of ionic species. If  $Z_i$  and  $n_i(r)$  are the valency and concentration of ionic species  $i$ ,

$$\rho(r) = \sum_{i=1}^N Z_i e n_i(r) \quad (2.4)$$

and

$$n_i = n_i^0 \exp (-Z_i e \psi / kT) \quad (2.5)$$

where  $n_i^0$  is the bulk concentration of ionic species  $i$ .

Combination of eq. (2.3) and eq. (2.4) leads to the Poisson-Boltzmann (PB) equation,

$$\frac{d^2 \psi(r)}{dr^2} = - \frac{1}{\epsilon_0 \epsilon_r} \sum_{i=1}^N Z_i e n_i^0 \exp (-Z_i e \psi(r) / kT) \quad (2.6)$$

Assuming the boundary conditions  $\psi(r) = \psi_0$  when  $r=0$  and  $\psi(r)=0$ ,  $d\psi/dr=0$  when  $r=\infty$ , then<sup>12</sup>

$$\psi(r) = \frac{2kT}{Ze} \ln \frac{1 + \gamma \exp (-\kappa r)}{1 - \gamma \exp (-\kappa r)} \quad (2.7)$$

where

$$\gamma = \frac{\exp [Z e \psi_0 / 2kT - 1]}{\exp [Z e \psi_0 / 2kT + 1]} \quad (2.8)$$

and

$$\kappa = \left( \frac{2 e^2 N_A c Z^2}{\epsilon_0 \epsilon_r kT} \right)^{1/2} \quad (2.9)$$

If the Debye-Hückel approximation is made,  $Z e \psi_0 / 2kT \ll 1$ , then,

$$\psi(r) \approx \psi_0 \exp (-\kappa r) \quad (2.10)$$

For a spherical double layer, the PB equation becomes

$$\nabla^2 \psi(r) = \frac{1}{r^2} \frac{d}{dr} \left( r^2 \frac{d\psi(r)}{dr} \right) \quad (2.11)$$

Loeb et al<sup>13</sup> point out that, unlike the case for a planar interface, there is no analytical solution to eq. (2.11). However, by using the Debye-Hückel approximation, the following expression is obtained,

$$\psi(r) = \psi_o \frac{a}{r} \exp(-\kappa(r-a)) \quad (2.12)$$

where  $a$  is the particle radius and  $r$  is the distance from its centre.

Loeb et al describe a numerical method for the solution of eq. (2.11) which does not require the Debye-Hückel approximation to be made.

In practice, the presence of adsorbed counterions at the surface means that  $\psi_o$  in the above equations should be replaced by the potential at the Stern plane,  $\psi_d$ . The Stern plane may be considered as the boundary between the dispersion medium and the particle, along with any adsorbed material.

### 2.2.2 THE INTERACTION BETWEEN TWO APPROACHING PARTICLES'

#### DOUBLE LAYERS

The electrostatic repulsion term,  $V_R$ , between two approaching particles in an aqueous electrolyte medium arises through the interaction

between their double layers as they approach. Hogg et al<sup>14</sup> have derived a general expression for the interaction between two particles using the Debye-Hückel approximation for identical particles, their expression is

$$V_R (H) = 2 \pi \epsilon_o \epsilon_r a \psi_d^2 \ln (1 + \exp (-\kappa H)) \quad (2.13)$$

For large particle separations, Reerink and Overbeek<sup>15</sup> obtain

$$V_R (H) = \frac{32 \pi \epsilon_o \epsilon_r a k^2 T^2 \gamma^2}{e^2 Z^2} \exp (-\kappa H) \quad (2.14)$$

for which the Debye-Hückel approximation is not made.

### 2.2.3 THE INTERPRETATION OF STABILITY FROM DLVO PLOTS

The above approximations for the interaction energy between two colloidal particles provide a powerful tool for interpreting and predicting colloid stability. For example, consider a dilute dispersion of spherical particles of radius 100nm in an aqueous solution of  $8 \times 10^{-3} \text{ mol dm}^{-3}$  potassium chloride at room temperature, with  $A_{pp} = 2 \times 10^{-19} \text{ J}$  and  $A_{mm} = 0.4 \times 10^{-19} \text{ J}^2$ . Figure 2.1 shows the DLVO plots obtained using eqs. (2.1, 2.2 and 2.14) as a function of the Stern potential,  $\psi_d$ .

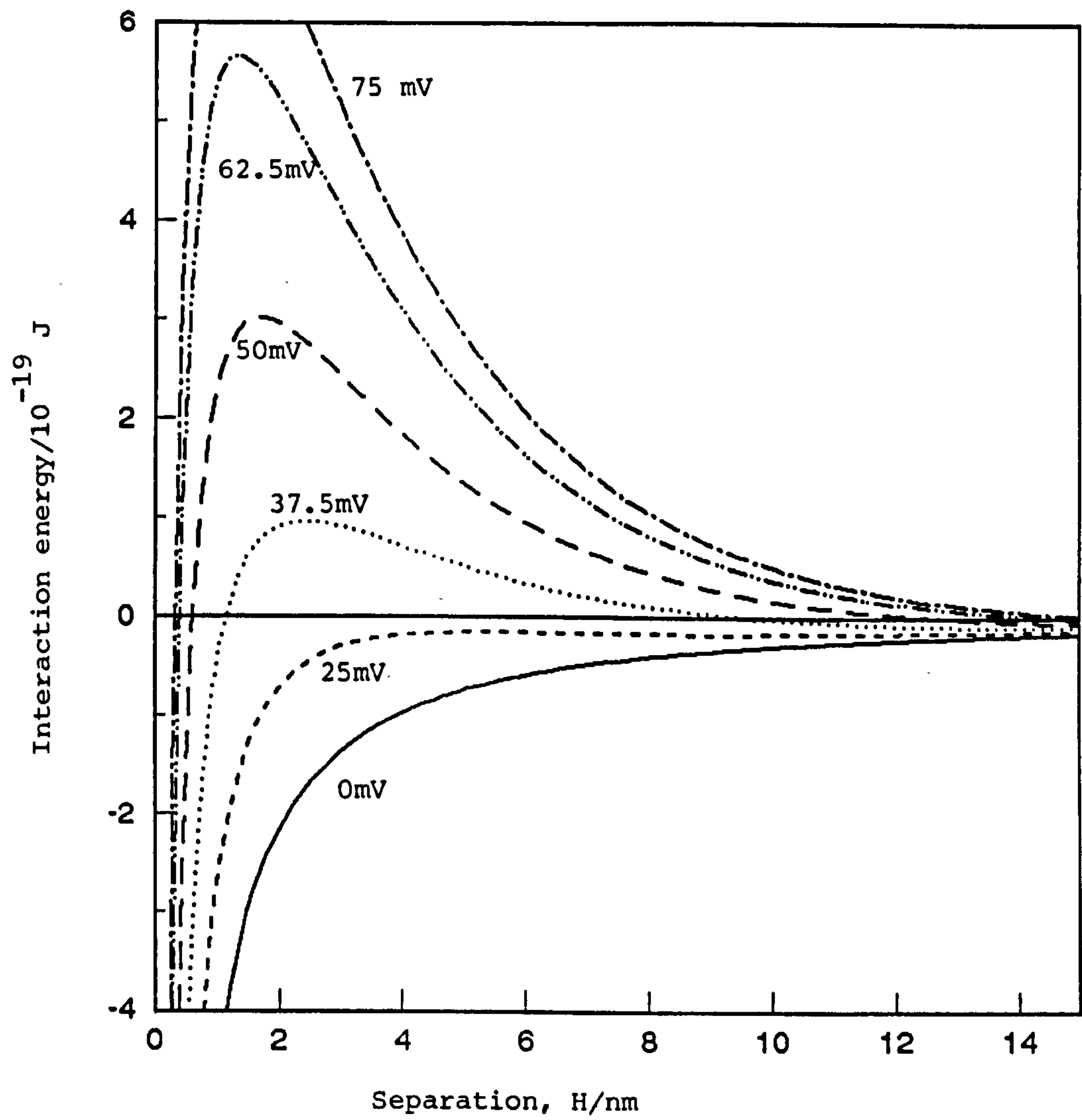


Figure 2.1: DLVO plots for a typical dispersion of charge stabilised particles in aqueous electrolyte solution as a function of Stern potential. (After ref. 2).

DLVO-type plots are usually quantitatively interpreted in terms of the kinetic stability of the dispersion by use of the "stability ratio", W. Smoluchowski<sup>16</sup> has shown that the rate of disappearance of primary (discrete) particles during the early stages of the coagulation process obeys

$$-\frac{dN}{dt} = k N_0^2 \quad (2.15)$$

where  $N_0$  is the initial number concentration of primary particles.

$K$  is the rate constant for the process and depends on the diffusion

of the particles, i.e.  $k = k_0 = 8 \pi D R$ , where  $D$  is the diffusion

coefficient of the particle and  $R$  is the so-called "collision radius"<sup>17</sup>.

In the presence of a suitable energy barrier,  $k$  is reduced to an

amount  $k_0/W$ . The higher  $W$ , the more kinetically stable the

dispersion.  $W$  can be related to the total interaction energy,

$$V_T(H) = V_A(H) + V_R(H), \text{ by }^6$$

$$W = 2a \int_0^{\infty} \exp\left(\frac{V_T(H)}{kT}\right) dH / (H + 2a)^2 \quad (2.16)$$

$W$  can be determined experimentally by turbidity measurements.

Assuming the necessary parameters are known in order to create the

interaction energy plot for a given system, the validity of the

DLVO theory applied to the system under investigation can be tested.



## 2.3 NON-AQUEOUS COLLOIDAL DISPERSIONS

In considering non-aqueous colloidal dispersions, two features need to be assessed. Firstly, the application of DLVO theory to such systems, with particular regard to the electrostatic repulsion interaction term. Secondly, the mechanisms governing the generation of charge/potential at the solid/liquid interface.

### 2.3.1 APPLICATION OF DLVO THEORY

The calculation of the repulsive interaction between two colloidal particles, where electrostatic forces are the only repulsive forces present, relies on a knowledge of  $\kappa$  (eq. (2.9)), which, in turn, requires that the concentration of electrolyte in the bulk phase be known. Non-aqueous dispersions can be (broadly) classified into two groups: semi-polar and non-polar. In semi-polar media, such as ethanol, electrolytes can be fully ionised up to high enough concentrations to make the application of the double layer theory thus far explained reliable. However, in non-polar media ( $\epsilon_r < 10$ ) electrolytes will tend to be poorly soluble and weakly dissociated. The concentration of dissociated species will be poorly defined, making estimation of  $\kappa$  very difficult. The low concentration of electrolyte (typically  $\sim 10^{-9} \text{ mol dm}^{-3}$ ) in media of low permittivity also results in  $\kappa$  being very small. Hence, the double layer thickness



will be very large, of the order of hundreds of nanometers. Consequently,  $\psi_0$  can be approximated to  $\psi_d$ . The slow decay of the potential away from the particle results in a low repulsion force, but which extends a long way into solution. In the limit of  $\kappa \rightarrow 0$ , the potential,  $\psi(r)$ , decays as

$$\psi(r) = \frac{Q}{4\pi \epsilon_0 \epsilon_r r} \quad (2.17)$$

which is Coulomb's law, where  $Q$  is the Coulombic charge on the particle. Eq. (2.17) is useful for calculating  $\psi_0$  when the electrolyte concentration in the bulk medium is ill-defined<sup>18</sup>. Due to the low permittivity of the bulk medium, the capacitance of the double layer will be low, hence small surface charges can yield large potentials. For example, a 200nm diameter particle in a medium where  $\epsilon_r = 4$ , possessing 14 elementary charge units on its surface, will have a surface potential of  $\sim 50$  mV.

However, one of the main assumptions used in the double layer theory is that the charge on a particle is uniformly distributed. Clearly, as the number of surface charges becomes small, the validity of this assumption must be questioned. Assuming the validity of eq. (2.17), then the repulsive interaction energy term for two particles in a non-polar dispersion is (cf. eq. (2.14))

$$V_R(H) = \frac{Q^2}{4 \pi \epsilon_o \epsilon_r (H + 2a)} = \frac{4 \pi \epsilon_o \epsilon_r a^2 \psi_o^2}{H + 2a} \quad (2.18)$$

From eq. (2.18), it can be seen that the repulsion term decays much more slowly than for a typical aqueous dispersion. One consequence is that the attractive energy term is less dominant in describing the overall interaction energy between particles. For example, figure 2.2 shows the interaction energy between two spherical particles, of radius 100 nm, for different surface potentials and Hamaker constants. For  $H > 100$  nm, the attraction term does not significantly contribute to the total interaction. It can be concluded<sup>18</sup> that, as long as electrostatic repulsion and van der Waals attraction are the only interactions operative, a precise knowledge of the latter is not necessary. Since  $V_A(H) \propto a$  and  $V_R(H) \propto a^2$ , larger particles in non-polar media are easier to electrostatically stabilise than smaller ones.

Although the simple Coulombic description for the repulsion term between two particles in a non-polar medium is useful, it does assume that the particles behave as point charges and, that as they approach, their surface charge remains constant. Parfitt et al<sup>19</sup> have derived the potential energy of charged colloidal particles in non-polar media by application of Maxwell's equations for the potential energy of charged conducting spheres. They consider two cases: (i) the surface charge remains constant as the particles

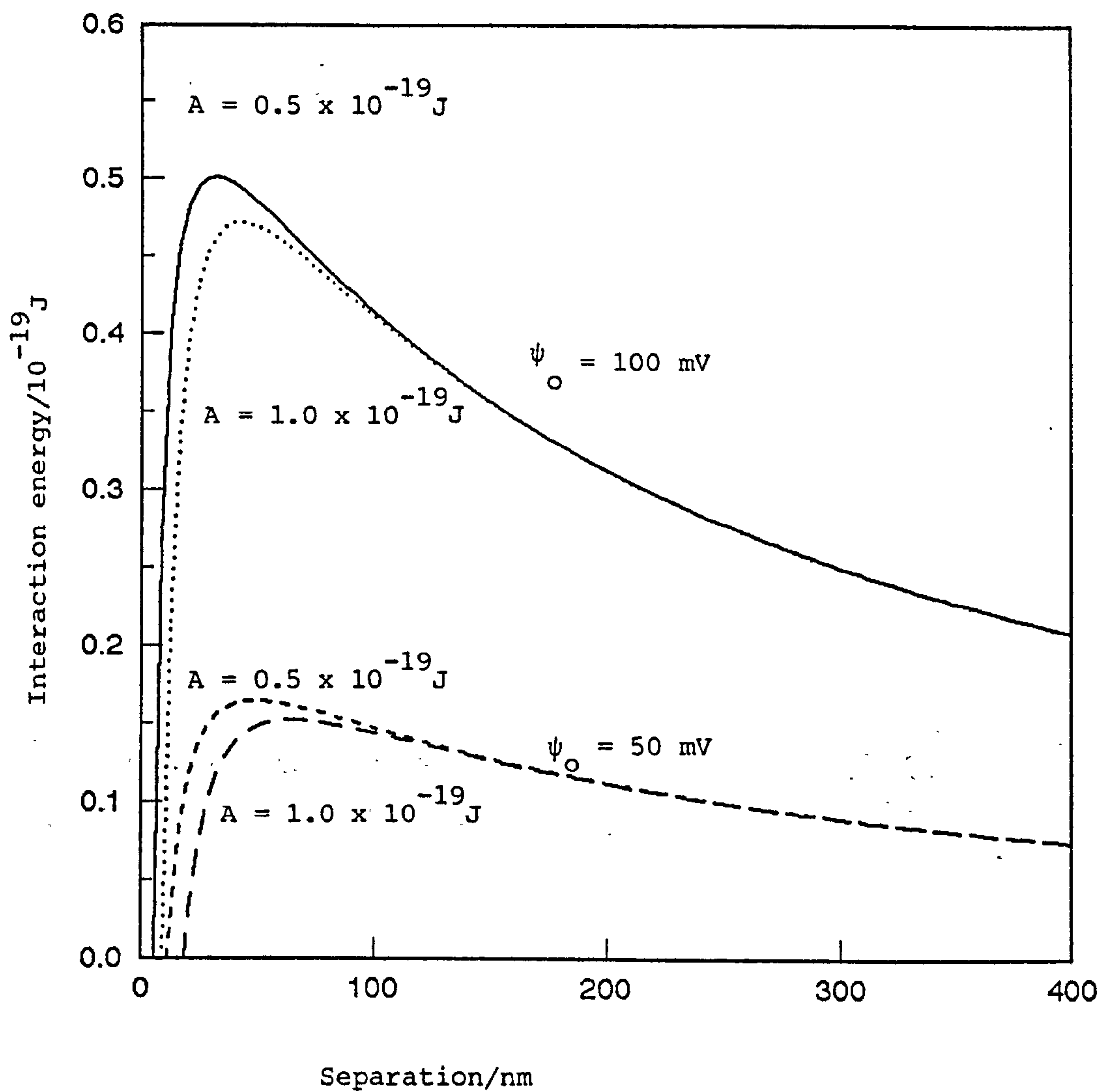


Figure 2.2: Interaction energy between two spherical particles ( $r = 100$ nm) dispersed in a low polarity medium ( $\epsilon_r = 2$ ) for different surface potentials,  $\psi_o$ , and Hamaker constants,  $A$ . (After ref. 18)

approach, and (ii) the surface potential remains constant as the particles approach. They show that discrepancies occur between this model and the simple Coulombic one, especially at small particle separations, with the constant potential model showing most deviation. The total electrostatic energy of a system of  $N$  charges,  $W_T$ , is

$$W_T = \frac{1}{2} \sum_{i=0}^N Q_i \psi_i \quad (2.19)$$

where  $Q_i$  is the magnitude of charge  $i$  and  $\psi_i$  is the potential at the location of this charge due to the remaining charges. For two conducting spheres in a dielectric medium, eq. (2.19) yields two expressions for the energy between the spheres. The first,  $W_c$ , is expressed in terms of the charge and the second,  $W_p$ , in terms of the potential, viz:

$$W_c = \frac{1}{8 \pi \epsilon_0 \epsilon_r} \left[ p_{11} Q_1^2 + 2p_{12} Q_1 Q_2 + p_{22} Q_2^2 \right] \quad (2.20)$$

$$W_p = 2 \pi \epsilon_0 \epsilon_r \left[ q_{11} \psi_1^2 + 2q_{12} \psi_1 \psi_2 + q_{22} \psi_2^2 \right] \quad (2.21)$$

$p_{11}$  and  $q_{11}$  etc. are functions of distance and sphere radius, where

$$q_{11} = a_1 (1 - \zeta^2) \sum_{n=0}^{\infty} \frac{\alpha^n}{1 - \zeta^2 \alpha^{2n}} \quad (2.22)$$

$$q_{12} = \frac{-a_1 a_2}{c} (1 - \alpha^2) \sum_{n=0}^{\infty} \frac{\alpha^n}{1 - \alpha^{2n+2}} \quad (2.23)$$

$$q_{22} = a_2 (1 - \eta^2) \sum_{n=0}^{\infty} \frac{\alpha^n}{1 - \eta^2 \alpha^{2n}} \quad (2.24)$$

$$\zeta^2 = (a_1 + a_2 \alpha)^2 / C^2 \quad (2.25)$$

$$\eta^2 = (a_2 + a_1 \alpha)^2 / C^2 \quad (2.26)$$

$$\alpha = k - \sqrt{k^2 - 1} \quad (2.27)$$

$$k = (1 - m^2 - n^2) / 2mn, \quad m = a_1 / C, \quad n = a_2 / C \quad (2.28)$$

$$\text{and } p_{11} = q_{22} / \Delta, \quad p_{22} = q_{11} / \Delta \text{ and } p_{12} = -q_{12} / \Delta \quad (2.29)$$

$$\text{where } \Delta = q_{11} q_{22} - q_{12}^2$$

$a_1$  and  $a_2$  are the radii of the two spheres and  $C$  is their centre-centre separation. The interaction between the two spheres for constant charge,  $V_{R(C)}$  is

$$V_{R(C)} = W_c - \frac{Q_1^2}{8 \pi a_1 \epsilon_0 \epsilon_r} - \frac{Q_2^2}{8 \pi a_2 \epsilon_0 \epsilon_r} \quad (2.30)$$

$$= W_c - 2 \pi \psi_1^2 a_1 \epsilon_o \epsilon_r - 2 \pi \psi_2^2 a_2 \epsilon_o \epsilon_r \quad (2.31)$$

For constant potential, the interaction,  $V_{R(P)}$ , is

$$V_{R(P)} = - (W_p - 2 \pi \psi_1^2 a_1 \epsilon_o \epsilon_r - 2 \pi \psi_2^2 a_2 \epsilon_o \epsilon_r) \quad (2.32)$$

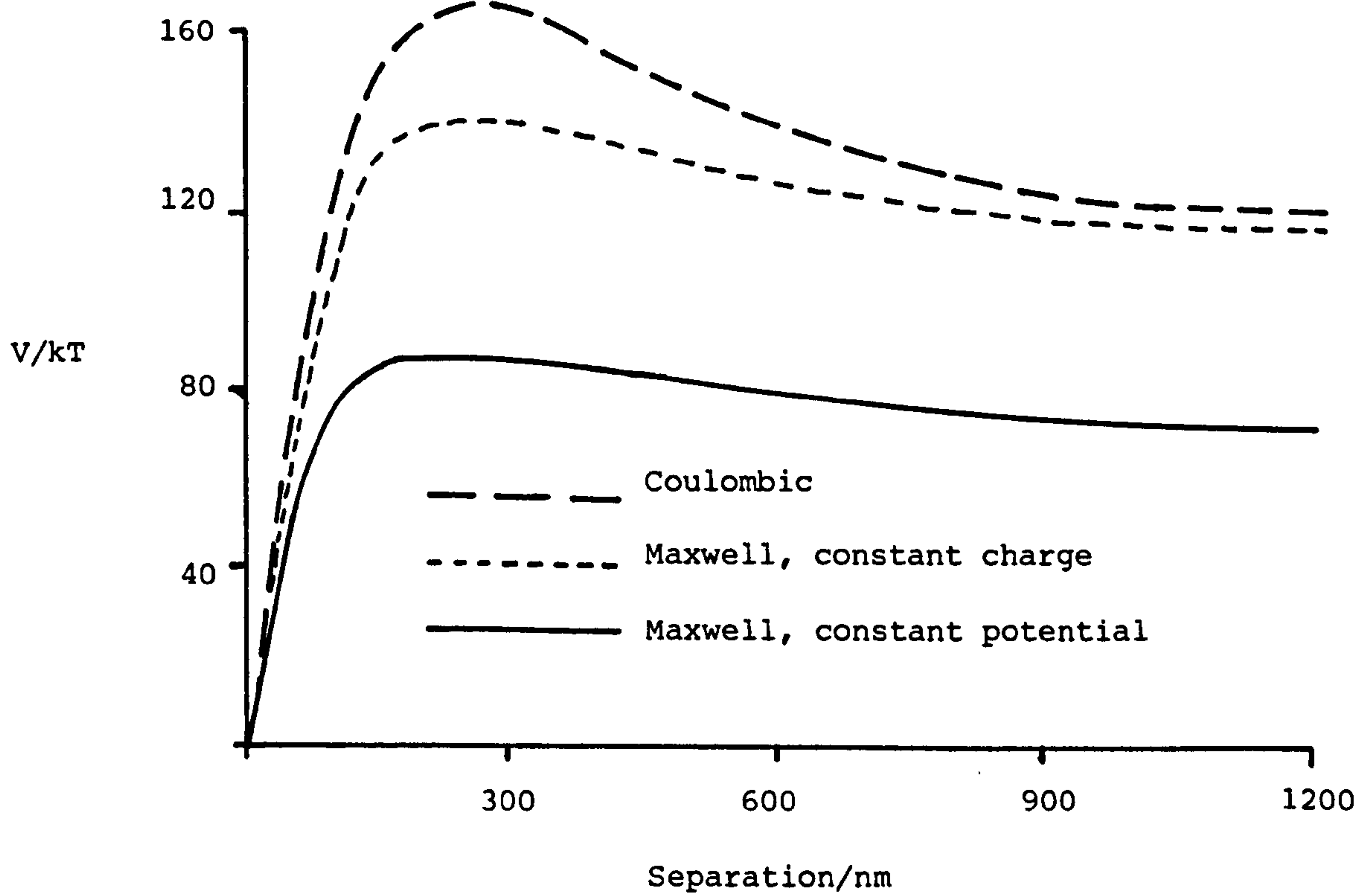
Figure 2.3 shows a comparison of interaction energy curves using eqns. (2.19, 2.31 and 2.32).

The use of Coulomb's law should be limited to large separations. Since the interaction potential for both methods only consider pairwise interactions, application is limited to dilute systems.

In a study of the surface potential and rate of coagulation of rutile ( $TiO_2$ ) particles dispersed in "Aerosol-OT"/p-xylene, McGown and Parfitt<sup>20</sup> conclude that DLVO theory can adequately predict the relationship between the stability ratio and the surface potential of the particles.

### 2.3.2 ORIGIN OF SURFACE CHARGE IN NON-POLAR DISPERSIONS

In his review in 1931, Soyenkoff<sup>21</sup> describes a wide variety of non-polar dispersions and addresses the question of interfacial charge. He states that a number of workers had observed electrophoresis



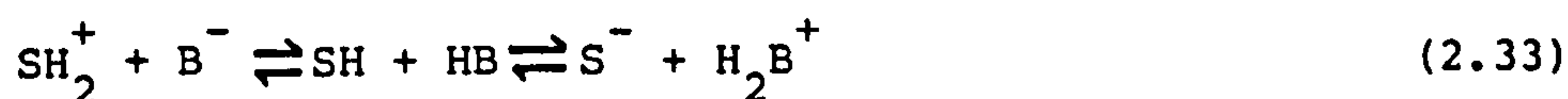
**Figure 2.3:** Representative interaction energy profiles for a charge-stabilised dispersion in a non-polar medium based upon Coulomb's law, Maxwell's law for constant charge and Maxwell's law for constant potential. After ref. 19.



(see § 2.4) of sols dispersed in benzene. For example, nickel carbonate dispersions in benzene were found to have a negative charge (Hatschek and Thorne, 1925). Pochettino (1911) found that dispersions of phenanthrene in benzene were negatively charged, whereas in carbon disulphide, they possessed a positive charge. Soyenkoff concluded, however, that electrical forces are not important for the stability of non-polar dispersions. Lyklema<sup>29</sup> states that the origin of charge in these types of system are likely to arise from two mechanisms,

- (i) the dissociation of surface groups
- and (ii) adsorption of ionic surfactants.

The ion most responsible for charge transfer between the dispersion medium and the particle is the proton. If SH and HB represent the solid and solution respectively, then we may write



where the equilibrium of eq. (2.33) lies to the left or right of centre, depending on the relative Brønsted acidities of the solid and solution. This concept has been supported by Verwey<sup>22</sup> who, in a series of electroosmotic experiments, found that the magnitude of the charge developed by various inorganic oxides ( $\text{SiO}_2$ ,  $\text{TiO}_2$  and  $\text{ZrO}_2$ ) in various media (water, acetone, ethanol and methanol) depends on the relative Brønsted acidity of each component. He found the following order of



acidities :  $\text{SiO}_2 > \text{TiO}_2 > \text{ZrO}_2$  and water < acetone < ethanol. McGown et al<sup>23</sup> have found that rutile ( $\text{TiO}_2$ ) is positively charged in xylene, as it is in n-butanol<sup>24</sup> and heptanol<sup>25</sup>, but it is negatively charged in butylamine<sup>24</sup> and nitrobenzene<sup>25</sup>.

The adsorption of ionic surfactants has been studied with regard to the charge thereby developed on the particle. For example,  $\text{SiO}_2$  and  $\text{CaCO}_3$  particles dispersed in xylene both become positively charged when sodium 1,2 - bis (2-ethyl hexyl oxycarbonyl)-1-ethane sulphate ("Aerosol OT" or  $\text{Na(AOT)}$ ) is added to the dispersion<sup>26, 27 28</sup>, whereas various carbon blacks dispersed in xylene in the presence of AOT acquire a negative charge<sup>26,20,29</sup>. Studies of the effects of the presence of various divalent AOT salts ( $\text{Mg(AOT)}_2$ ,  $\text{Ca(AOT)}_2$  and  $\text{Ba(AOT)}_2$ ) on carbon blacks dispersed in benzene<sup>30</sup> show that a positive surface charge is developed. The above observations have been attributed to preferential adsorption of ionic species. In the case of  $\text{Na(AOT)}$ , it is thought that the  $\text{AOT}^-$  ion preferentially adsorbs onto the carbon black surface. For the divalent surfactants, Parreira<sup>30</sup> proposes that the surfactant exists in solution as



although the equilibrium lies strongly to the left in the above equation. Preferential adsorption of the  $\text{M}^+(\text{AOT})$  results in an overall positive surface charge. The association constant,  $K_A$ , for ionogenic compounds in non-polar media can be estimated by the Bjerrum-Fuoss<sup>31</sup>

equation,

$$K_A = \frac{4 \pi \delta^3 N_A}{3000} \exp (e^2 Z_+ Z_- / 4 \pi \epsilon_0 \epsilon_r \delta kT) \quad (2.35)$$

where  $Z_{\pm}$  are the ionic charges and  $\delta$  is the distance of closest approach between paired ions, being of the order of  $0.5 \text{ nm}^{29}$ .

In the last few years, an alternative explanation for the origin of charge in non-polar dispersions has been put forward by Labib et al<sup>32,33,34</sup>. They relate the charge developed at a solid/liquid interface to the difference between the Lewis basicities of the solid and the liquid. The Lewis basicity of a substance can be expressed in terms of its "electron donicity", which is conveniently related to the enthalpy of reaction between the substance of interest and a reference Lewis acid,  $\text{Sb Cl}_5$ , in 1,2-dichloroethane. A more basic material, in the Lewis sense, will yield a higher enthalpy of reaction. The enthalpy of reaction, in  $\text{kcal mol}^{-1}$ , is termed the donor number,  $D_N$ . For organic solvents which are more acidic than 1,2-dichloroethane, in the Lewis sense, an analogous "acceptor number",  $A_N$  is used, based upon the relative  $^{31}\text{P}$  n.m.r. shift caused by dissolving triethylphosphine oxide ( $\text{Et}_3 \text{P=O}$ , a reference Lewis base) in the solvent. A scale of 0 to 100 is used for  $A_N$ , 0 for the shift observed for  $\text{Et}_3 \text{P=O}$  in hexane and 100 for 1:1  $\text{Et}_3 \text{P=O} : \text{Sb Cl}_5$  in 1,2-dichloroethane. Schmid<sup>35</sup> has compiled a list of  $A_N$  and  $D_N$  values for a wide selection

of organic solvents. Labib et al ascribe the surface charge developed in a non-aqueous dispersion to the donor-acceptor interaction between the solid and the liquid. For an electron transfer occurring at the interface, two reactions are possible,



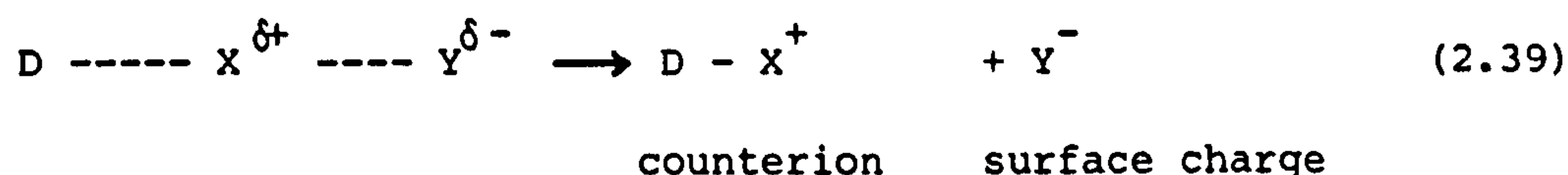
where P and S represent the particle and the solvent respectively.

For ion transfer occurring at the interface, the possible interactions are



donor molecule    surface bond    donor-acceptor complex

then, if the donicity of D: is high enough,



Similarly, for a strong acceptor liquid, A,



Labib and Williams<sup>32</sup> have obtained a series of zeta potentials (see § 2.4) for a variety of solids in liquids with varying  $D_N$  values, which support the mechanism proposed above for the origin of charge in non-polar dispersions.

One factor known to be responsible for affecting the surface charge characteristics in non-polar dispersions, which has not been discussed yet, is the presence of trace amounts of water in the dispersion. The previously mentioned studies by Damerell and Urbanic<sup>26</sup> of the effects of various surfactants on carbon black dispersed in xylene included some crude electrophoresis experiments, performed in a U-tube. It was found that the addition of water to the carbon black (i.e. the powder was exposed to water vapour) resulted in a decrease in the stability of the dispersion. A similar phenomenon was found for Damerell and Mattson's<sup>27</sup> study of calcium carbonate dispersions in surfactant/xylene solutions. However, van der Waarden's<sup>36</sup> study of carbon black dispersions in various hydrocarbon/surfactant solutions showed no difference in stability in the absence or presence of trace water. Chessick et al<sup>37</sup> found that, in non-polar media, water enhances the flocculation ability of polar solids. This was attributed to the formation of water layers around the solids. When two water layers come into contact, flocculation of the solid is promoted. In a study of the adsorption by pure rutile of various straight chain, primary alcohols from solution in p-xylene,

Parfitt and Wiltshire<sup>38</sup> found that the less polar the alcohol, the less adsorbed and that adsorbed water on the rutile caused a significant reduction in the amount of alcohol adsorbed. In McGown et al's<sup>23</sup> study of the relationship between surface potential and the stability of the rutile/AOT/p-xylene system, comparisons were made with similar studies on  $\alpha$ -alumina, carbon black and copper phthalocyanine. Addition of water to the rutile system was found to be very important in governing the sign and magnitude of the surface potential. The potential was found to go through a maximum with increasing water concentration. Increasing the concentration of AOT was found to decrease the potential. In a discussion of McGown et al's work, Fowkes<sup>39</sup> states that sometimes the addition of water can reduce the electrophoretic mobility (from which McGown et al determined the surface potentials) to zero, but without affecting streaming potentials. Fowkes interprets this as a surface conductivity effect. Parfitt<sup>40</sup> states that the positive charge found is attributable to water adsorbed at the rutile/solution interface. Very dry rutile/p-xylene/AOT possesses a negative charge on the solid's surface. Micale et al's<sup>25</sup> investigation into the mechanism by which trace water affects the dispersion characteristics of rutile in organic media showed that the particle charge exhibited a maximum at a fractional water coverage of  $\sim 0.05$ . Griot<sup>41</sup> investigated the influence of adsorbed water on the electrophoretic mobility of anatase ( $\text{TiO}_2$ ) in non-aqueous media. It was found that



removal of physisorbed water increased the mobility and that removal of chemisorbed water decreased the mobility. Recently, Malbrel and Somasundaran<sup>42</sup> have investigated the effect of water on the stability of alumina dispersed in cyclohexane in the presence of AOT. The settling rate and the adsorbed amount of water were determined as a function of the initial water concentration in the bulk phase. They found that on increasing the water content, the settling rate firstly decreases sharply from  $\sim 10 \mu\text{m s}^{-1}$  to  $\sim 0.1 \mu\text{m s}^{-1}$  and then, as the water content increases further, the settling rate increases. This corresponds to the general trend found by McGown et al<sup>23</sup> for the surface charge. Adsorption isotherms for water onto the rutile were found to be analogous to those found for vapour adsorption onto solid substrates. The increase in stability was found to coincide with the vapour adsorption-type stage (i.e. up to a monolayer coverage). As the water adsorbed more readily (analogous to the condensation of vapour onto a solid), the stability decreased. In other recent studies, Goodwin et al<sup>43, 44</sup> investigated the effect of trace water on non-aqueous silica dispersions, using a butanol/dodecane mixture as the dispersion medium (20% butanol by mass). They found that in the presence of a non-ionic surfactant, hexaethylene oxide dodecyl ether, the electrophoretic mobility of the silica particles changed sign from positive to negative at a water

concentration in the bulk of 0.05% by mass. However, the conclusions made from these studies did not account for the possibility of preferential adsorption of butanol<sup>45</sup>, which could affect the surface potential of the particles.

van der Hoeven and Lyklema<sup>46</sup> have investigated the stability of sodium tri(poly)phosphate dispersed in a non-ionic surfactant, "Plurafac RA30", in the presence of an anionic surfactant, dodecyl benzenesulphonic acid. They found that the zeta potential decreased from ~ 100 mV to ~ 10 mV as the water content rose from ~ 0.1% to 2.5% by mass, whilst the conductivity of the system remained constant. They attribute this behaviour to water adsorbing onto the particle. Van der Hoeven<sup>47</sup> has also performed streaming potential measurements to confirm that the zeta potential drop as obtained by electrophoresis is not due to a surface conductance effect.

## 2.4 ESTIMATION OF SURFACE POTENTIAL AND CHARGE VIA ELECTROKINETIC PHENOMENA

### 2.4.1 ELECTROKINETIC PHENOMENA

The stability of an electrostatically stabilised colloidal dispersion has been related to the potential at the Stern plane,  $\psi_d$ .

For a system with  $\kappa a \rightarrow 0$ , this potential is very close to the potential at the surface of the particle,  $\psi_o$ . The various equations describing the electrostatic repulsion between two particles require a knowledge of  $\psi_d$ .  $\psi_d$  is considered to be very similar to the zeta potential,  $\zeta$ , which is the experimentally determined quantity used for the estimation of  $\psi_o$  or  $\psi_d$ . Measurement of  $\zeta$  relies on the exploitation of one of four electrokinetic phenomena<sup>48</sup>:

- (i) ELECTROOSMOSIS: Reuss<sup>49</sup> found that when an electric field was applied to a V-tube containing water and a quartz sand plug, the water began to move. Without the sand plug, the water did not move; i.e. the applied electric field caused the liquid to flow relative to a stationary, charged surface. This phenomenon is electroosmosis.
- (ii) ELECTROPHORESIS: Reuss also found that application of an electric field to clay particles in water resulted in displacement of the particles whilst the liquid remained at rest. This describes an example of electrophoresis, which can be defined as the motion of a charged surface plus attached material (e.g. Stern layer) relative to a stationary liquid due to an applied electric field.



- (iii) STREAMING POTENTIAL: Quincke<sup>50</sup> demonstrated that the opposite of electroosmosis exists; i.e. the generation of an electric potential due to the flow of liquid over a stationary charged surface. The potential generated is termed the streaming potential.
- (iv) SEDIMENTATION POTENTIAL: This is the potential developed when charged particles move relative to a stationary liquid and is the opposite of electrophoresis.

All these electrokinetic phenomena are related to the zeta potential. The most widely used phenomenon for measuring  $\zeta$  for colloidal dispersions is electrophoresis.

#### 2.4.2 DETERMINATION OF ZETA POTENTIAL USING ELECTROPHORESIS

When a charged particle dispersed in a suitable liquid medium is exposed to an externally applied electric field,  $E$ , it will accelerate to a uniform terminal velocity,  $V_E$ . The velocity will depend on the relative magnitudes of the electric and viscous drag forces acting on the particle. It is found that  $V_E$  is proportional to  $E$ ,

$$V_E = \mu E \quad (2.41)$$

where  $\mu$  is the proportionality constant, referred to as the electrophoretic mobility. It is necessary to find a relationship

between  $\mu$  and  $\zeta$ . Von Smoluchowski<sup>51</sup> has considered the analogous case for electroosmosis. His treatment assumes<sup>52</sup>:

- (i) the usual hydrodynamic equations for the flow of a viscous fluid hold for both the bulk liquid and that in the double layer,
- (ii) the motion is "stream line" and slow enough for "inertia terms" to be neglected,
- (iii) the applied field is simply superimposed on that due to the electrical double layer,
- (iv) the double layer thickness is small compared with the radius of the solid, i.e.  $\kappa a \gg 1$ .

The following relationship is thus obtained :

$$\mu_o = \frac{-\epsilon_o \epsilon_r \zeta}{\eta} \quad (2.42)$$

where  $\mu_o$  is the electroosmotic mobility of the liquid and  $\eta$  its viscosity.  $\mu_o$  is simply the negative of  $\mu$ . Hence, the electrophoretic mobility can be related to  $\zeta$  by

$$\mu = \frac{\epsilon_o \epsilon_r \zeta}{\eta} \quad (2.43)$$

which is Smoluchowski's equation for electrophoresis. However, Debye and Hückel<sup>53</sup> criticise Smoluchowski's treatment, stating that

$$\mu = \frac{\epsilon_0 \epsilon_r \zeta}{\eta} \cdot K \quad (2.44)$$

where  $K$  is a constant, dependent upon particle shape, and can only be obtained by detailed hydrodynamic analysis of the motion of the liquid. For a spherical particle, Hückel<sup>54</sup> obtains

$$\mu = \frac{\epsilon_0 \epsilon_r \zeta}{\eta} \cdot \frac{2}{3} \quad (2.45)$$

and does not make assumption (iv) above.

In an attempt to explain the difference between eq. (2.43) and eq. (2.44), Henry<sup>52</sup> reexamined Debye and Hückel's analysis and found that the applied potential gradient is unperturbed by the presence of the particle. This is only true when the particle and the medium have the same electrical conductivity. If  $\lambda_p$  and  $\lambda_l$  are specific electrical conductivities of the solid particle and liquid medium, respectively, then, when  $\lambda_p \ll \lambda_l$ , the field will be distorted such that the lines of force near the particle run tangentially to it. Smoluchowski assumed that the field was uniform and parallel to the particle surface. Two extreme cases for the effect of a non-conducting particle on the

applied field can be envisaged. One where  $\kappa a \ll 1$  (Hückel limit) and the other where  $\kappa a \gg 1$  (Smoluchowski limit). Figure 2.4 depicts these two cases.

Henry addressed the problem of the deformation of the field by the particle and attempted to derive an expression with eqs. (2.43) and (2.45) as the limiting cases (for  $\kappa a \rightarrow \infty$  and  $\kappa a \rightarrow 0$ , respectively), i.e.

$$\mu = \frac{2}{3} \cdot \frac{\epsilon_0 \epsilon_r \zeta}{\eta} f(\kappa a) \quad (2.46)$$

For a spherical particle, Henry obtained for  $\kappa a < 1$ ,

$$f(\kappa a) = 1 + \frac{(\kappa a)^2}{16} - \frac{5(\kappa a)^3}{48} - \frac{(\kappa a)^4}{96} + \frac{(\kappa a)^5}{96} - \left[ \frac{(\kappa a)^4}{8} - \frac{(\kappa a)^6}{96} \right] \exp(\kappa a) \int_{\infty}^{\kappa a} \frac{1}{t \exp t} dt \quad (2.47)$$

and for  $\kappa a > 1$ ,

$$f(\kappa a) = \frac{3}{2} - \frac{9}{2\kappa a} + \frac{75}{2(\kappa a)^2} - \frac{330}{(\kappa a)^3} \quad (2.48)$$

$f(\kappa a)$  is referred to as Henry's function. The shape of the field around the particle depends on both  $\kappa a$  and the ratio of  $\lambda_p$  to  $\lambda_1$ . If  $\lambda_p / \lambda_1 = K'$ , Henry derives a more general expression

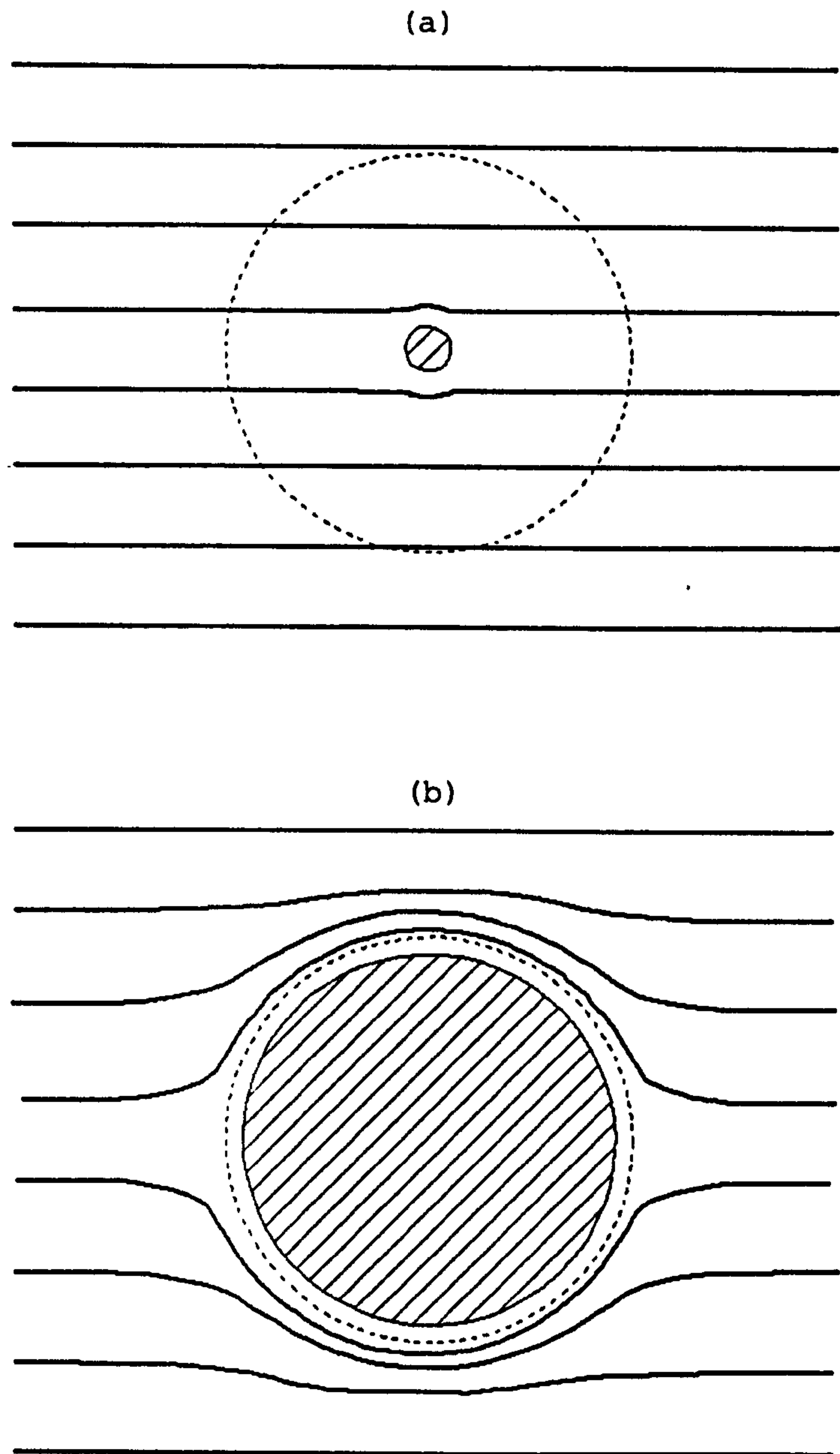


Figure 2.4: Distortion of electric field around a non-conducting particle for (a)  $\kappa a \ll 1$  (Hückel) and (b)  $\kappa a \gg 1$  (Smoluchowski). Broken circle represents a distance  $1/\kappa$  from the particle surface.

$$\mu = \frac{2}{3} \cdot \frac{\epsilon_0 \epsilon_r \zeta}{\eta} \cdot F(\kappa a, K') \quad (2.49)$$

where  $F(\kappa a, K') = 1 + 2 \lambda [f(\kappa a) - 1]$  and  $\lambda = (1 - K')/(2 + K')$ .

Figure 2.5 illustrates how  $F(\kappa a, K')$  changes with  $\kappa a$ . Henry has also considered the effects of surface conductance<sup>55</sup>. If  $\lambda_s$  is the surface conductivity of the particle,  $\lambda$ , as defined above, becomes

$$\lambda = \frac{1 + K' - 2\lambda'_s/a}{2 + K' + 2\lambda'_s/a} \quad (2.50)$$

where  $\lambda'_s = \lambda_s/\lambda_1$  and  $a$  is the radius of the particle. However, as Hunter<sup>56</sup> points out, surface conductance effects are normally taken into account by consideration of the mobility of the double layer ions.

Henry's treatment (above) assumes that the external field can be superimposed on that due to the particle, and that the latter can be described by a linear form of the Poisson-Boltzmann equation. As he points out, however<sup>55</sup>, the application of an external field will distort the field around the particle. The spherical symmetry of the field around the particle is thereby lost and a reduction in the electrophoretic velocity occurs. This is the "relaxation effect". As the particle moves, the surrounding ionic atmosphere must reform by motion of the double layer ions, hence the extent of the relaxation

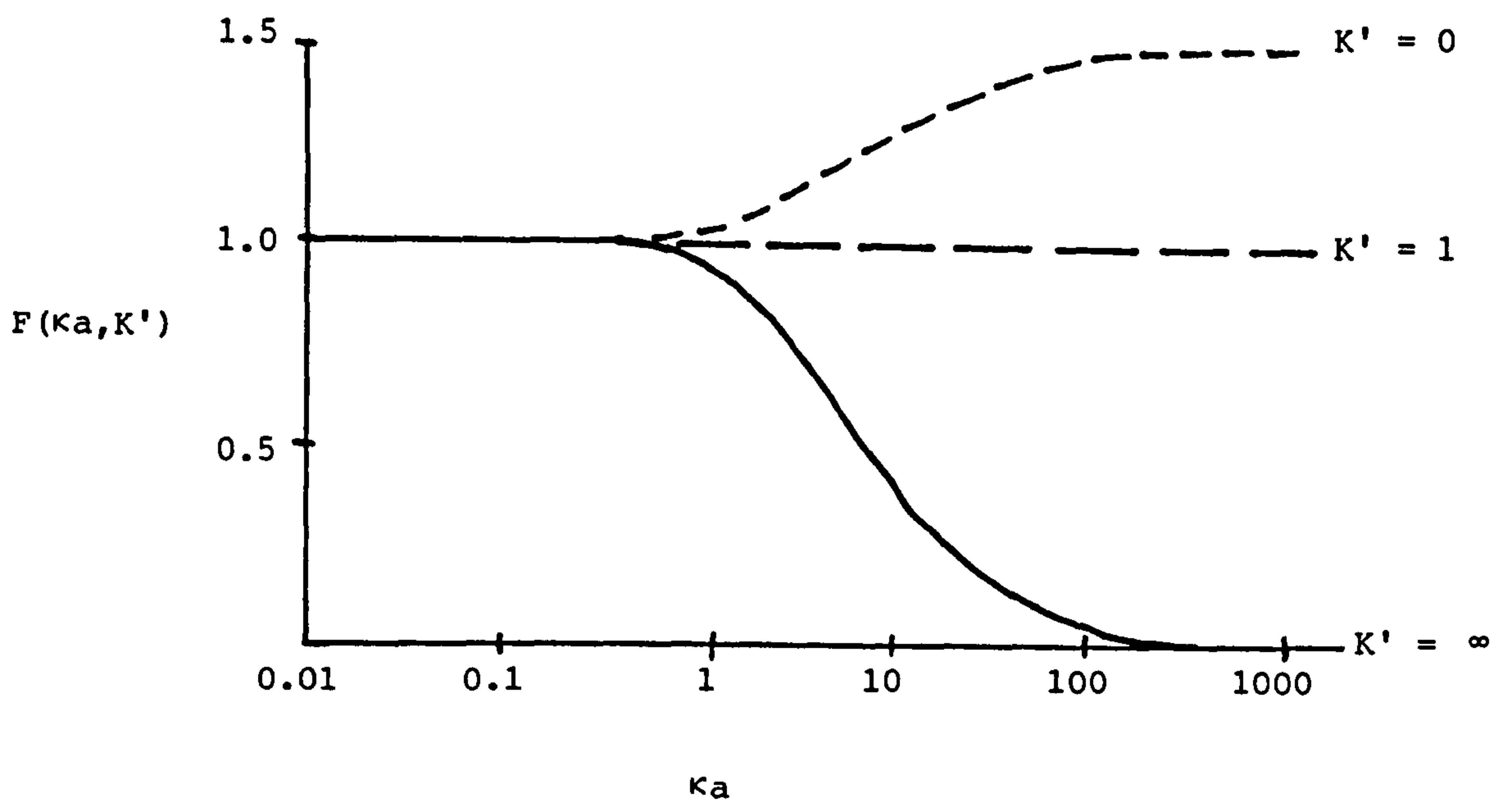


Figure 2.5: Variation of  $F(\kappa a, K')$  with  $\kappa a$  for a spherical particle with  $K' = 0, 1$  and  $\infty$ .



will depend on the mobility and charge of the counterions. Hermans<sup>57</sup> states that for  $\kappa a \approx 1$ , the relative error in relating  $\mu$  to  $\zeta$  via eq. (2.49) is of the order of 100% and for  $\kappa a \ll 1$ , eq. (2.43) fails. He states further that the situation of  $\kappa a \ll 1$  rarely occurs in practice since the particle diameter is usually so large that no electrolyte concentration exists where  $\kappa a \ll 1$ , except when the solvent is non-polar.

When the electric field is applied to a dispersion, the particles will migrate in one direction, but the counterions in the double layer will move in the opposite direction, resulting in a further reduction of the electrophoretic velocity. This is termed "electrophoretic retardation". In the region where  $\kappa a \approx 1$  (i.e. the transition between the Hückel and Smoluchowski limits), the force due to the applied field, the force due to frictional resistance by the liquid and the forces due to electrophoretic retardation and relaxation effects are of the same order of magnitude<sup>48</sup>. Henry's equation needs to be modified to account for these factors. Overbeek<sup>58</sup> derives a set of equations based upon the following assumptions :

- (i) the particle is a rigid sphere with a surface potential  $\zeta$  and a charge  $ne$  spread out uniformly over the surface,
- (ii) the particle is embedded in a solution of a  $(Z_+ - Z_-)$  valent electrolyte,

- (iii) the particle is surrounded by a Gouy-Chapman type diffuse double layer that is deformed by electrophoresis.

The general equation obtained is

$$\begin{aligned} \mu = \frac{2}{3} \frac{\epsilon_0 \epsilon_r \zeta}{\eta} & \left[ f_1(\kappa a) - (z_- - z_+) \frac{e\zeta}{kT} f_2(\kappa a) + \right. \\ & (z_+ z_-) \left( \frac{e\zeta}{kT} \right)^2 g_1(\kappa a) - (z_+^2 - z_+ z_- + z_-^2) \left( \frac{e\zeta}{kT} \right)^2 g_2(\kappa a) \\ & + (z_- - z_+)^2 \left( \frac{e\zeta}{kT} \right)^2 g_3(\kappa a) - \frac{z_+^{\rho_+} + z_-^{\rho_-}}{(z_+ + z_-) e} \cdot \frac{\epsilon_0 \epsilon_r kT}{\eta} \\ & \left. \cdot \frac{2}{3} \left( \frac{e\zeta}{kT} \right)^2 f_4(\kappa a) \right] \end{aligned} \quad (2.51)$$

$f_1(\kappa a)$  is Henry's function. Overbeek lists a table of  $f_1(\kappa a)$ ,  $f_2(\kappa a)$ ,  $f_3(\kappa a)$  and  $f_4(\kappa a)$  for  $\kappa a = 0.01$  to  $1000$ . For a symmetrical electrolyte, eq. (2.51) is approximated to

$$\begin{aligned} \mu \approx \frac{2}{3} \cdot \frac{\epsilon_0 \epsilon_r \zeta}{\eta} & \left[ f_1(\kappa a) - z^2 \left( \frac{e\zeta}{kT} \right)^2 f_3(\kappa a) - \right. \\ & \left. \frac{\rho_+ + \rho_-}{2e} \cdot \frac{2}{3} \frac{\epsilon_0 \epsilon_r kT}{\eta e} \left( \frac{e\zeta}{kT} \right)^2 f_4(\kappa a) \right] \end{aligned} \quad (2.52)$$

and for an unsymmetrical electrolyte

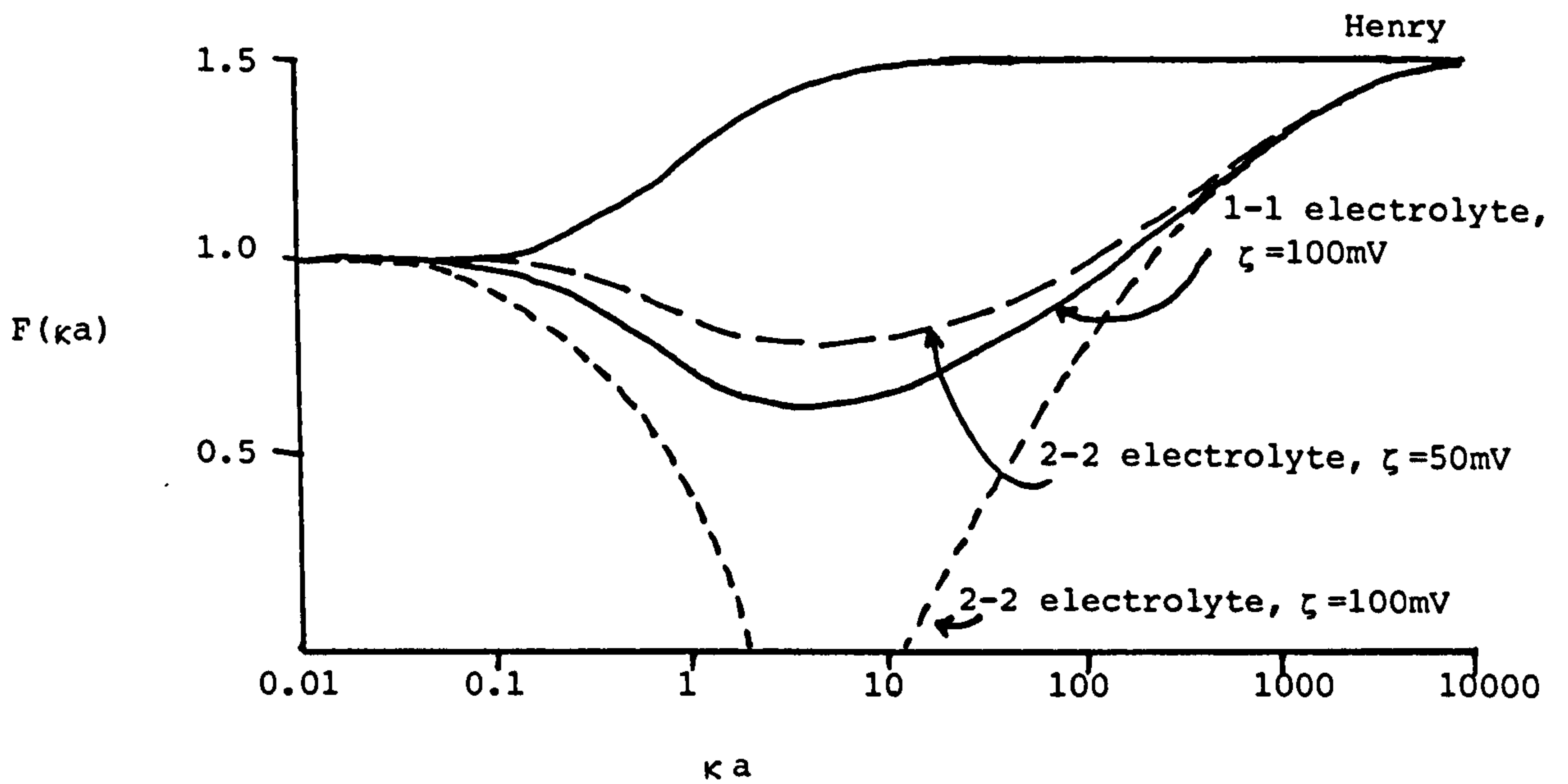
$$\begin{aligned} \mu \approx \frac{2}{3} \cdot \frac{\epsilon_0 \epsilon_r \zeta}{\eta} & \left[ f_1(\kappa a) - (z_- - z_+) \frac{e\zeta}{kT} f_2(\kappa a) - \right. \\ & \left. \frac{z_+^{\rho_+} + z_-^{\rho_-}}{(z_+ + z_-) e} \cdot \frac{2}{3} \cdot \frac{\epsilon_0 \epsilon_r kT}{\eta e} \left( \frac{e\zeta}{kT} \right)^2 f_4(\kappa a) \right] \end{aligned} \quad (2.53)$$

where  $\rho_{\pm}$  are the friction coefficients of the ions. Figure 2.6 illustrates how Overbeek's treatment modifies Henry's function type plots.

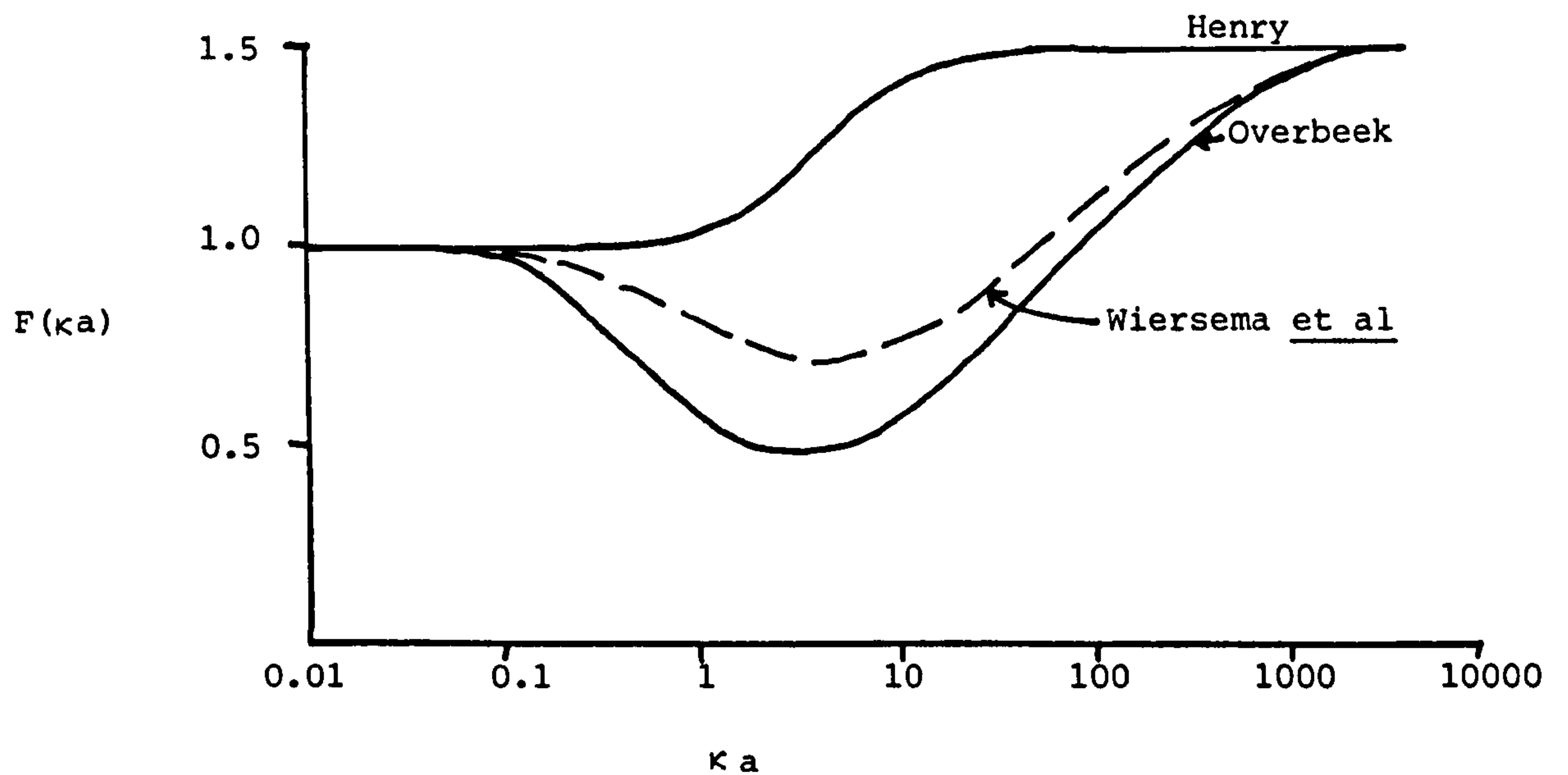
Overbeek states that for  $\zeta < 25$  mV, the correction factor for Henry's function never exceeds ~3%.

With the advent of superior computational power, Wiersema et al<sup>59</sup> were able to solve the appropriate differential equations behind Overbeek's method without approximations. The overall conclusion is that Overbeek's original method over-compensates for relaxation effects. Figure 2.7 represents graphically Henry's function, Overbeek's approximate equations and the exact numerical solution.

Although Wiersema et al have provided a means of relating  $\mu$  to  $\zeta$  which accounts for relaxation effects, the process of determining  $\zeta$ , given  $\mu$ , is rather tedious, involving the construction of a number of graphs. O'Brien and White<sup>60</sup> have developed an alternative method for relating  $\mu$  to  $\zeta$  based upon the same set of equations used by Wiersema et al. Exact consideration of the balance between the forces acting on the particle lead to a set of coupled non-linear partial differential equations. Solution of these is quite formidable, so O'Brien and White obtain a set of linear equations by assuming that the applied field is small compared with the fields that occur in the double layer, the latter being of the order of  $\zeta\kappa$ .



**Figure 2.6:** Comparison between Henry's function and Overbeek's functions for relating  $\mu$  to  $\zeta$  as a function of  $\kappa a$ .



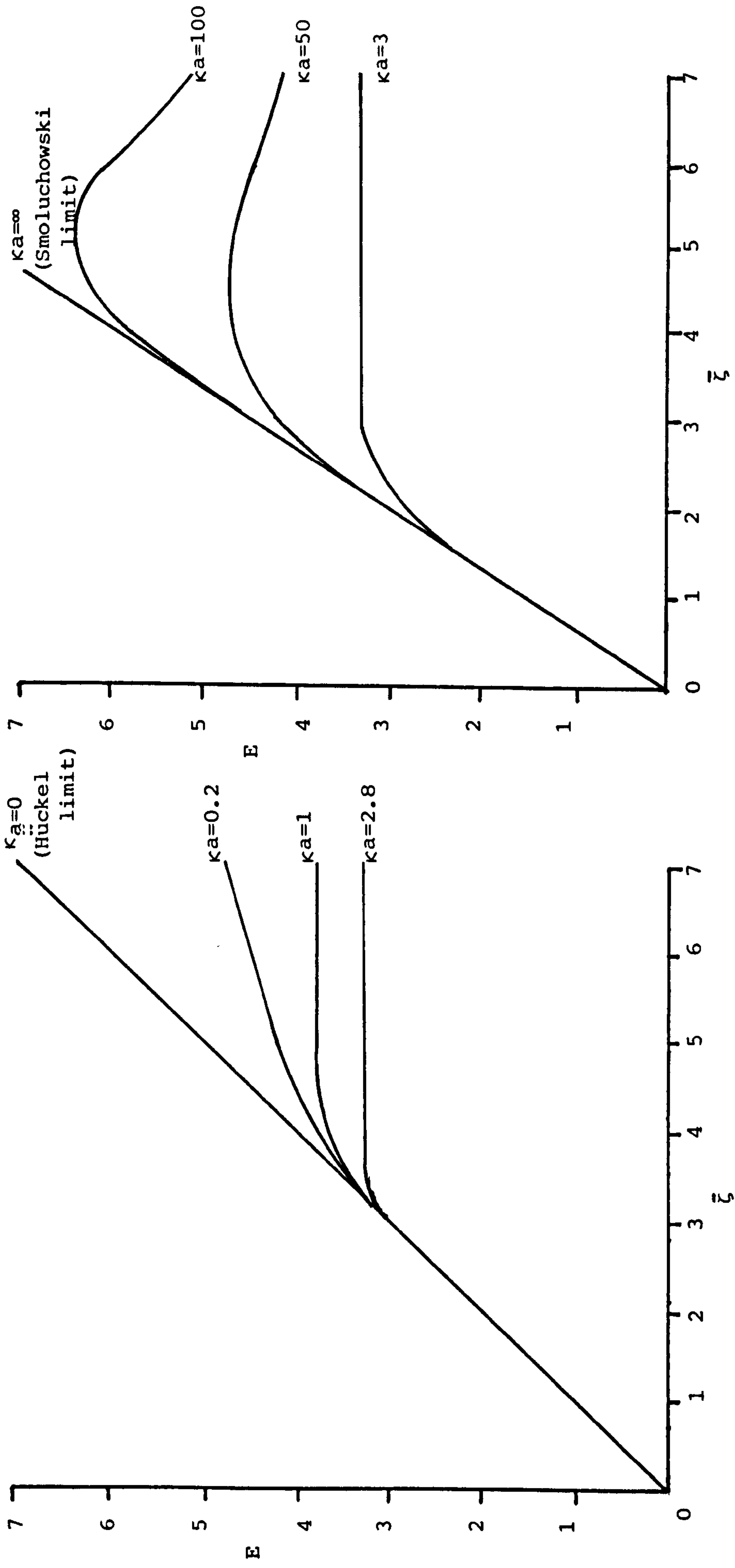
**Figure 2.7:** Comparison between Henry's function, Overbeek's approximate function and Wiersema et al's exact numerical solution for Overbeek's method, (1-1 electrolyte,  $\zeta = 125\text{mV}$ . See text for explanation).

O'Brien and White's method suggests that for  $\kappa a > 3$ , a maximum exists in the relationship between the reduced mobility,  $E$ ,  
 $(E = (3\eta e)/(2 \epsilon_0 \epsilon_r kT) \cdot \mu)$  and the reduced potential  $\bar{\zeta}$  ( $\bar{\zeta} = e\zeta/kT$ ).  
 Some illustrative examples are shown in figure 2.8.

The expected mobility maximum has been observed in some systems<sup>61</sup>. Levine and Bell<sup>62</sup> have tried to explain the mobility maximum using a model of the Stern layer where  $\zeta$  decreases as the surface potential rises, taking into account the discreteness of the charge. O'Brien and White's theory suggests that the mobility maximum originates from the dynamic behaviour of the diffuse part of the double layer.

More recently, Ohshima et al<sup>63, 64</sup> have derived a semi-empirical analytical formula for O'Brien and White's theory, which yields relative errors of less than 1% for  $\kappa a > 10$ . Their overall relation between  $E$  and  $\bar{\zeta}$  is

$$\begin{aligned}
 E = & \frac{3}{2} \bar{\zeta} - \frac{3F}{1+F} H + \frac{1}{\kappa a} \left[ -18 \left( t + \frac{t^3}{9} \right) K + \right. \\
 & \frac{15F}{1+F} \left( t + \frac{7t^2}{20} + \frac{t^3}{9} \right) - 6 (1 + 3m_+) (1 - \exp(-\bar{\zeta}/2)) G \\
 & \left. + \frac{12F}{(1+F)^2} H + \frac{9\bar{\zeta}}{1+F} (m_+ G + m_- H) - \frac{36F}{1+F} \left( m_+ G^2 + \frac{m_-}{1+F} H^2 \right) \right]
 \end{aligned}
 \tag{2.54}$$



**Figure 2.8:** Relationship between reduced mobility,  $E$ , and reduced potential,  $\bar{\zeta}$ , for various  $\kappa a$  according to O'Brien and White's method.

where

$$K = 1 - \frac{25}{3 (\kappa a + 10)} \exp \left( - \frac{\kappa a}{6 (\kappa a - 6)} \bar{\zeta} \right) \quad (2.55)$$

$$t = \tanh (\bar{\zeta} / 4) \quad (2.56)$$

$$F = \frac{2}{\kappa a} (1 + 3m_-) (\exp (\bar{\zeta}/2) - 1) \quad (2.57)$$

$$G = \ln ((1 + \exp (-\bar{\zeta}/2))/2) \quad (2.58)$$

$$H = \ln ((1 + \exp (\bar{\zeta}/2))/2) \quad (2.59)$$

$$m_{\pm} = \frac{2 \epsilon_o \epsilon_r kT}{3\eta Z^2 e^2} \cdot \lambda_{\pm} \quad (2.60)$$

and  $\lambda_{\pm}$  are the drag coefficients of the ionic species at infinite dilution.

## 2.5 CLASSICAL METHODS FOR THE DETERMINATION OF ELECTROPHORETIC MOBILITIES

A variety of techniques for studying the electrophoresis of particles have been developed, each designed to either overcome limitations of other methods or to enhance a particular feature. For example, moving boundary electrophoresis is widely used for the separation of proteins. However, it is unsuitable for studying



dilute colloidal dispersions. A description of the moving boundary method can be found elsewhere<sup>65</sup>.

The most useful technique for the study of dilute colloidal dispersions is microelectrophoresis, which is extensively used for studying biological systems, such as blood cells. The principle behind the method is straight forward. The dispersion to be studied is subjected to a well-defined electric field and the resultant velocity of the particles is determined by observing their motion through an ultramicroscope. Figure 2.9 illustrates the general features of a typical microelectrophoresis apparatus. The particles are observed flowing along a narrow, cylindrical glass capillary. However, due to surface charge on the glass walls, application of the electric field results in electroosmosis in a direction opposing that of the electrophoresis. Since the net displacement of fluid must be zero, in the centre of the capillary the return flow of liquid will increase the velocity of the particles moving due to the external electric field. It is found that the velocity distribution of the particles across the capillary is parabolic. At some distance from the wall, the fluid motion due to electroosmosis will be balanced by the return flow. At this point, the particle velocity will be the true electrophoretic velocity. This point is termed the stationary layer, the position of which can be easily calculated<sup>66</sup>. It is possible to coat the inside of the capillary with, for example, agarose, which prevents

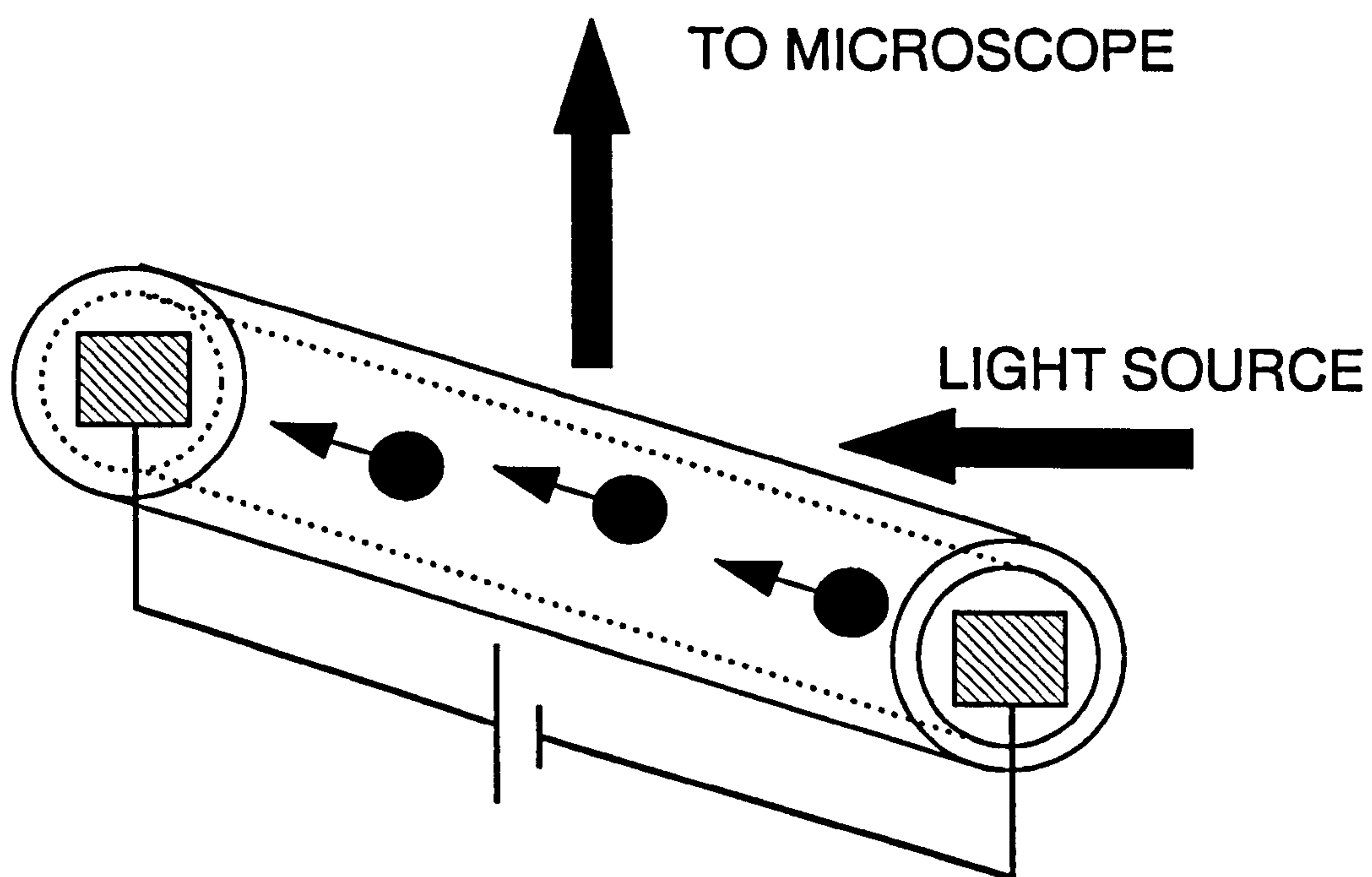


Figure 2.9: Schematic representation of a microelectrophoresis apparatus.

electroosmosis<sup>67</sup>. Hence, the velocity profile for the particles across the capillary becomes constant. Unfortunately, the material adsorbed onto the capillary wall may also adsorb onto the particles being studied.

The determination of particle velocity is made by timing how long a particle takes to traverse a known distance. A calibrated graticule in the microscope is typically used. During a typical experiment, a dozen or so particles may be observed. The length of the capillary is usually ~ 150 mm. The electrodes are usually constructed from platinum foil cylinders situated at either end of the capillary.

Determination of the electric field strength requires a knowledge of the effective electrode separation. This can be determined by using aqueous solutions of potassium chloride, for which the specific conductivity is known, as a calibrant. Bangham et al<sup>66</sup> claim that the effective electrode separation is very similar to their physical separation. Typical field strengths applied to aqueous samples are of the order of  $5 \text{ V cm}^{-1}$ , thus requiring ~ 75 V to be applied across the electrodes.

### 2.5.1 EXTENSION TO NON-AQUEOUS DISPERSIONS

The classical microelectrophoresis method described above has found wide application for the study of dispersions in aqueous electrolyte solutions. It is necessary now to consider the problems associated with extending the technique to non-aqueous dispersions. The problems arise through the low permittivity of the dispersion medium and the often low electrical conductivity of the samples. From eq. (2.45) it is clear that for a particle with a given zeta potential, the mobility in, say, 1, 4-dioxan ( $\epsilon_r \approx 2.3$ ;  $\eta = 1.1$  cP at room temperature) will be  $\sim 40$  times lower than in water ( $\epsilon_r \sim 80$ ;  $\eta \sim 0.9$  cP). Hence, higher voltages (of the order of kV) will need to be applied across the electrodes if suitable electrophoretic velocities are to be observed. Two problems may arise from the application of such voltages. Firstly, if the conductivity of the sample is less than that of the capillary wall (i.e.  $< 10^{-10}$  S cm $^{-1}$ ), a large proportion of the current may flow through the glass wall, rather than the sample. This will result in a poorly-defined, inhomogeneous field across the sample. Secondly, Joule heating of the sample may occur, which, if above a certain value, will result in a transition from laminar to turbulent flow, making determination of electrophoretic velocities difficult. This is an example of electrohydrodynamics. The threshold voltage,

$V_T$ , above which turbulence will occur is approximately

$$V_T = 30 \eta / (\rho \epsilon_0 \epsilon_r)^{\frac{1}{2}} \quad (2.61)$$

where  $\eta$ ,  $\rho$  and  $\epsilon_r$  are the viscosity, density and relative permittivity of the dispersion medium respectively<sup>68</sup>. Eq. (2.61) assumes unipolar conduction. For hydrocarbons,  $V_T$  is typically 200 - 300 V.

Another phenomenon that may be experienced with low conductivity, non-aqueous dispersions is that due to space charge effects. When an electric field is initially applied across a sample, the field across the sample will be uniform. As oppositely charged species begin to separate, a macroscopic, non-zero charge density distribution arises. This can lead to non-uniform, variable electric fields across the sample. When this occurs, the system is said to be "space charge limited". This problem is minimised by using very dilute samples and by inducing large electrophoretic velocities<sup>68</sup>. For particles with small surface charge, space charge effects are likely to be negligible.

Van der Minne and Hermanie<sup>69</sup> have described an apparatus based on the one described above that can be used for studying non-aqueous dispersions with low conductivity. They achieved this by

reducing the thickness of the capillary walls and by constructing the capillary from quartz rather than glass. The capillary itself is housed in an evacuable glass jacket. The risk of conductance through the capillary wall is thereby reduced. They found that field strengths of  $12 - 150 \text{ V cm}^{-1}$  were required to study the systems of interest (benzene/carbon black/calcium diisopropyl salicylate), corresponding to  $\sim 150$  to  $2000 \text{ V}$  across the electrodes. Hence, the possibility of electrohydrodynamic effects occurring with this design are high.

In order to minimise space charge effects, high field strengths are desirable. However, low voltages are required if electrohydrodynamic turbulence is to be avoided. Both can be achieved if the electrode separation is reduced from  $\sim 15 \text{ cm}$  to  $\sim 1$  to  $2 \text{ mm}$ . The problem then arises of how to determine the velocity of the particles moving between such narrow-gap electrodes. This is one of the reasons why the laser-Doppler methods described later are so useful.

A number of workers have studied electrophoresis in low conductivity liquids, but performed the experiments with very high field strengths, utilising voltages above  $V_T$  (eq. (2.61))<sup>70, 71</sup>.



One other important aspect to consider with non-aqueous electrophoresis studies is that of field-induced polarisation<sup>72</sup>. It is possible that application of an external electric field to a dispersion may result in polarisation of the particles, particularly in low permittivity media. As a result, the electrophoretic motion of the particles will be affected. If the polarisation of surface groups is ignored, the following equation is obtained.

$$V_E = \frac{2 \zeta \epsilon_0 \epsilon_1}{3\eta} \cdot E + \frac{4 \pi a^2 \epsilon_0 \epsilon_1}{3\eta} \left( \frac{\delta\epsilon}{\epsilon_2 + 2 \epsilon_1} \right) \nabla E^2 \quad (2.62)$$

where  $V_E$  is the electrophoretic velocity of a particle of radius  $a$ , relative permittivity  $\epsilon_1$ , dispersed in a liquid of viscosity  $\eta$  and relative permittivity  $\epsilon_2$ .  $\delta\epsilon$  is the difference between  $\epsilon_1$  and  $\epsilon_2$ .  $E$  is the applied field strength and  $\nabla E^2$  is the gradient of the square of the field strength, which is non-zero for an inhomogeneous field.

When  $\delta\epsilon$  or  $\nabla E^2$  are zero, eq. (2.62) reduces to Hückel's equation. The right hand velocity term is the dielectrophoretic velocity, which should be minimised by suitable design of the experiment. The absence of dielectrophoresis during an experiment can be confirmed by showing that the electrophoretic mobility is invariant with applied field strength and polarity. It will be seen later that the systems of interest for this study have  $\delta\epsilon \sim 0$ .



### 2.5.2 ELECTROPHORESIS IN ALTERNATING ELECTRIC FIELDS

So far, only the electrophoretic response of particles in a d.c. field has been considered. The use of a.c. fields is of importance since it can help reduce polarisation effects occurring at the electrode surfaces and allows for experiments of longer duration to be performed. Moreover, the use of alternating fields can reduce electroosmosis. This has been considered by Vorob'eva et al<sup>73</sup>. They show that using low frequency a.c. fields (0.5 - 10 Hz), the normal parabolic velocity profile across the cell is obtained. However, at higher frequencies (~ 30 Hz), the electroosmotic velocity away from the wall is reduced. This is illustrated in figure 2.10.

Above frequencies of ~ 1 Hz, manual determination of particle velocity clearly becomes difficult. Delatour and Hanss<sup>74</sup> have described a modification to the classical microelectrophoresis apparatus. Whilst an a.c. field is applied to a sample, the whole cell assembly is driven mechanically upwards. A photographic plate records the trace of the particles. Figure 2.11 illustrates the principle behind their method. However, in order to obtain suitably large amplitudes of oscillation, high field strengths need to be applied. It is interesting to note the similarity between this method and the PALS method described later, which also utilises a.c. sinusoidal fields of ~ 30 Hz.

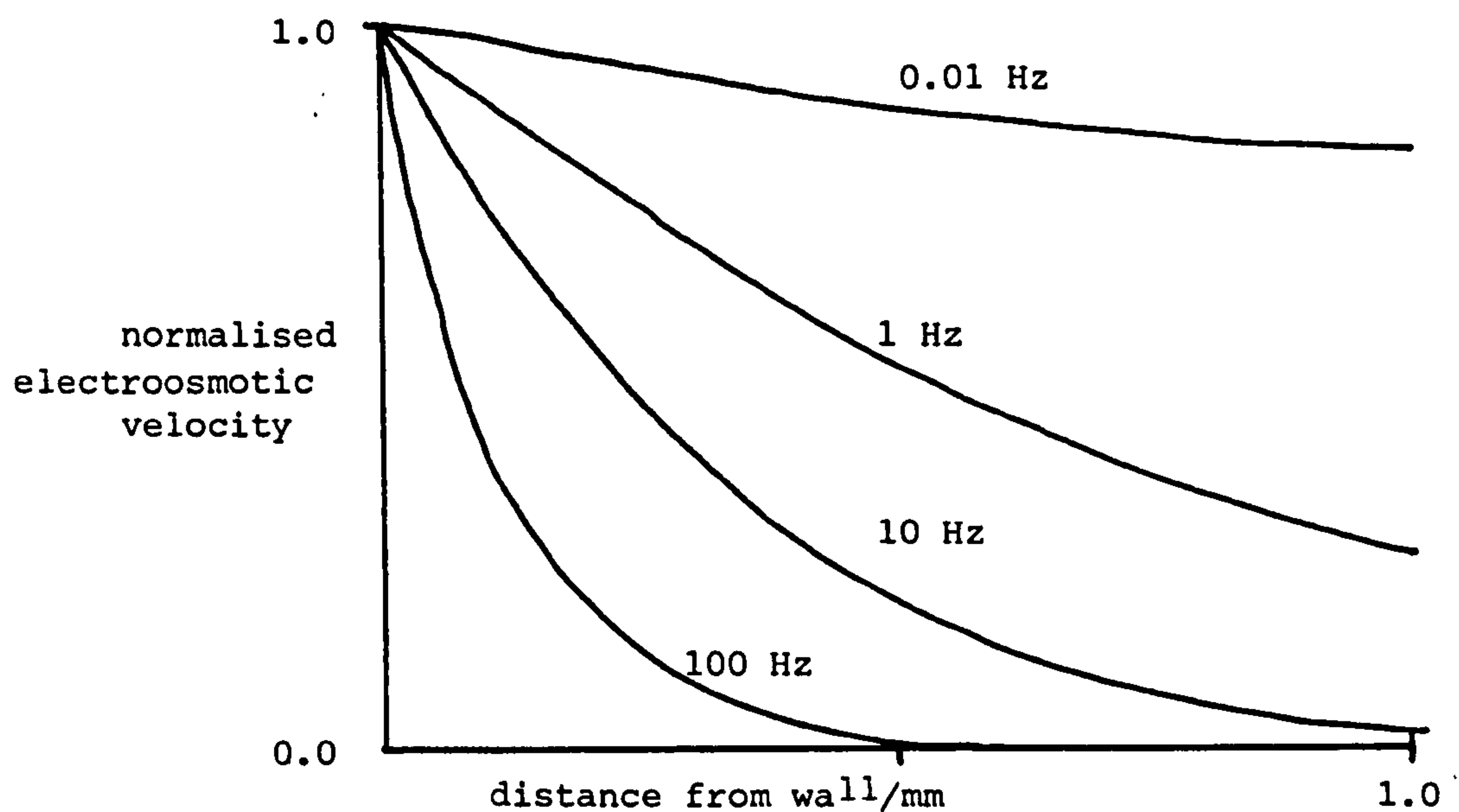


Figure 2.10: Variation in electroosmotic velocity as a function of distance from cell wall for different a.c. field frequencies. (After ref. 73).

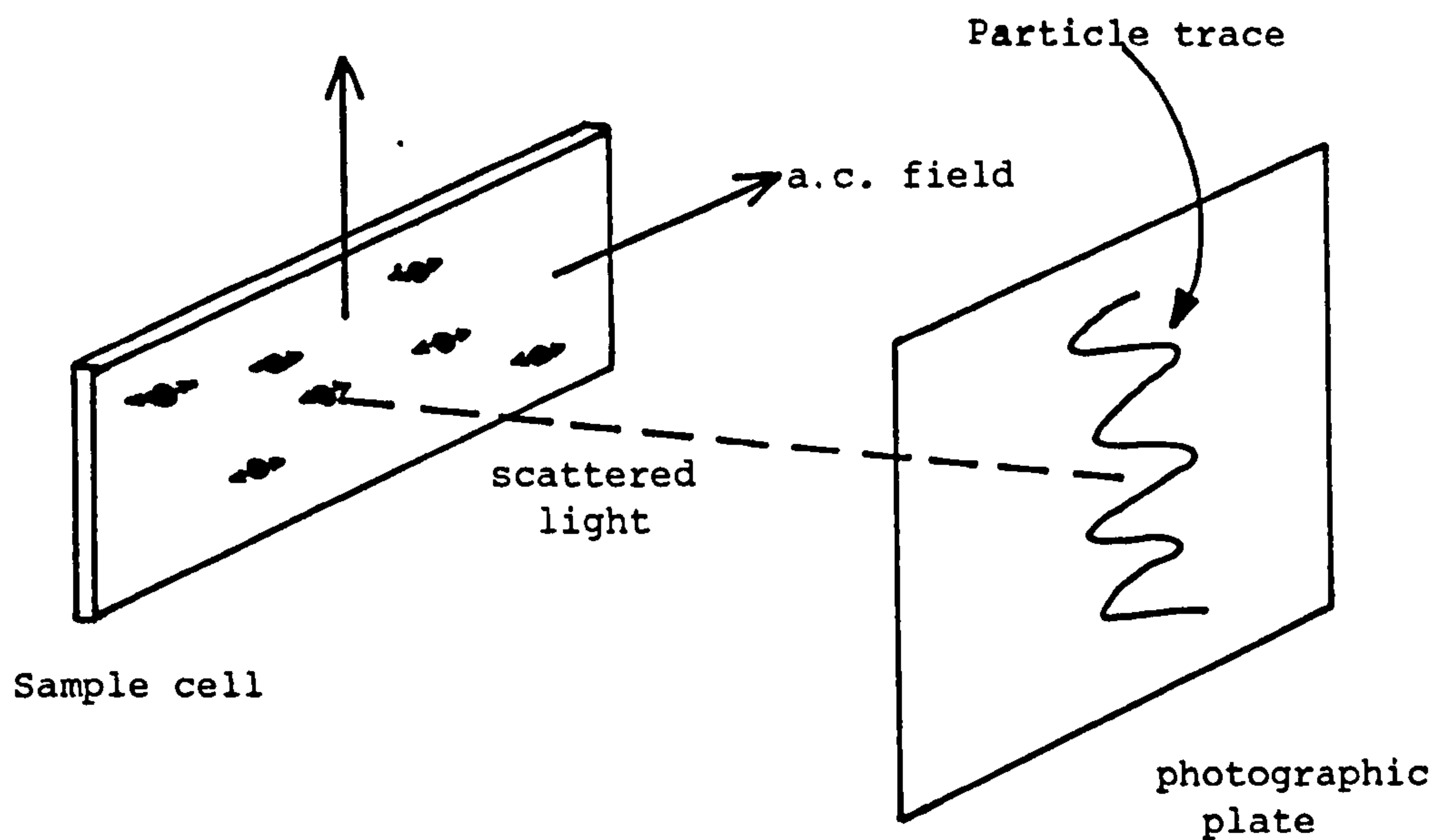


Figure 2.11: Schematic representation of Delatour and Hanss' microelectrophoresis apparatus using an a.c. field. (After ref. 74).

More recently, Gaigalas et al<sup>75</sup> have reported their investigations into the electrophoretic response of submicron particles to alternating electric fields. They claim that for sinusoidal fields below 50 Hz, the electrophoretic response is highly non-linear with field frequency. They used a laser-Doppler technique to determine velocity information. However, it is evident from their paper that they have misinterpreted the experimental data (see § 3.6.7). It will be seen, with the PALS equipment, that a linear electrophoretic response is obtained for sinusoidal fields below ~ 50 Hz.

## CHAPTER THREE

### LIGHT SCATTERING

#### 3.1 BASIC PRINCIPLES OF LIGHT SCATTERING<sup>1, 76</sup>

##### 3.1.1 SCATTERING FROM SMALL PARTICLES

Consider a small, isotropic spherical particle, whose diameter is less than  $\sim 1/20$  of the wavelength of a plane polarised, monochromatic light wave directed at the particle. At any given instant, the electric field of the light wave will induce an electric dipole moment in the particle, due to the motions of the electrons and, to a lesser extent, the protons in the particle. The oscillations of the electrons result in the generation of a secondary electromagnetic field, thereby radiating spherical light waves of the same frequency as the incident wave. For an unpolarised light source, the scattered light intensity,  $I$ , at an angle  $\theta$  to the incident light beam and a distance  $r$  from the particle, is given by

$$\frac{I r^2}{I_0} = \frac{8 \pi^4 \alpha^2}{\lambda^4} (1 + \cos^2 \theta) \quad (3.1)$$

where  $I_0$  is the incident light intensity (proportional to the square of the electric field amplitude,  $E_0$ ),  $\lambda$  is the wavelength of the

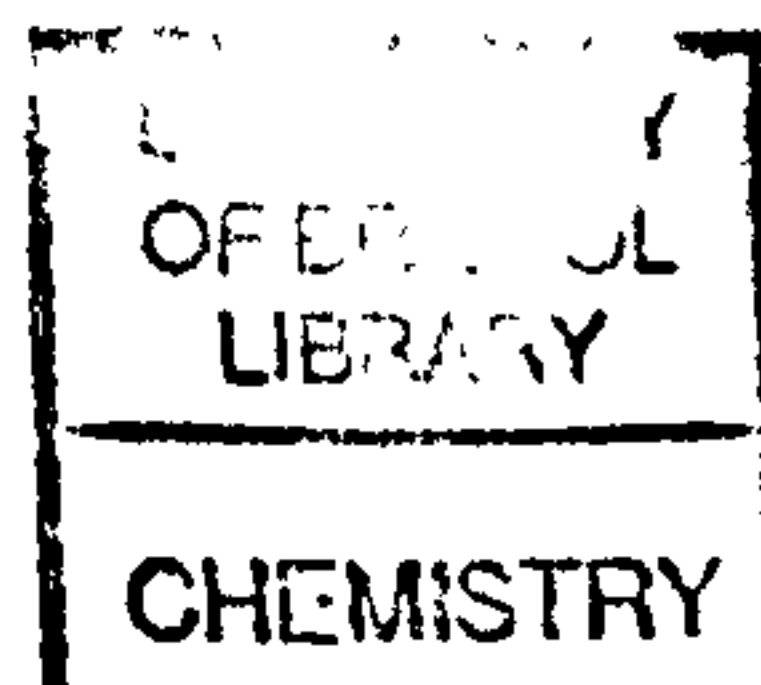
incident beam and  $\alpha$  is the polarisability of the particle.  $\alpha$  is given by

$$\alpha = 3 \epsilon_0 \left( \frac{n^2 - 1}{n^2 + 3} \right) V \quad (3.2)$$

where  $\epsilon_0$  is the permittivity of free space,  $V$  the volume of the particle and  $n$  the relative refractive index (i.e.  $n = n_1/n_0$ , where  $n_1$  is the refractive index of the particle and  $n_0$  that of the dispersion medium). Hence, it can be seen from eq. (3.2) that the scattering ability of a particle depends on the refractive index difference between it and the bulk phase. For an index-matched dispersion there will be very little scattering. Any scattering will be mainly due to thermal fluctuations causing local density changes, thereby changing the local refractive index.

### 3.1.2 SCATTERING FROM LARGE PARTICLES

The above discussion on the scattering of light from a small particle assumes that dipoles causing the secondary emission of radiation are all oscillating in phase with each other and, hence, the particle can be treated as a point source of scattered radiation. However, as the dimensions of the particle approach that of the wavelength of the incident radiation, interference effects will occur between scattered radiation originating from different locations within the particle. Destructive interference events are more probable between radiation scattered in the backwards direction.



Hence, the radiation envelope, described by eq. (3.1), loses its symmetry, becoming larger in the forward direction. For a very large particle with a large refractive index difference between itself and the dispersion medium, the radiation envelope becomes very complex, exhibiting angular dependent maxima and minima. This can be used for estimating the size of large particles.

### 3.2 SCATTERING FROM A COLLOIDAL DISPERSION

The above account of light scattering has only considered scattering from a single, stationary particle. It is necessary to introduce the effects of scattering from many particles moving in a liquid.

If a monochromatic, coherent, unidirectional light source (i.e. that of a laser) is directed at a colloidal crystal (i.e. a dispersion containing stationary particles arranged in a crystal-type lattice), a diffraction pattern will be obtained, analogous to that obtained from a molecular crystal placed in an X-ray beam. The diffraction pattern occurs as a result of interference events between light scattered from different particles.

If, however, the particles are randomly distributed, instead of obtaining a diffraction pattern containing well-defined maxima,



a random 'speckle' pattern is obtained. (See Everett<sup>1</sup> and Pike<sup>77</sup> for photographic examples of such speckle patterns). At a given fixed point in space, the scattered light intensity will be random but constant. If, instead of stationary, randomly distributed particles, the dispersion consists of randomly diffusing particles, the speckle pattern will change randomly. The intensity of scattered light at a fixed point in space will now fluctuate. Clearly, slowly diffusing particles will result in slow intensity fluctuations, whereas rapidly diffusing particles will result in fast intensity fluctuations. Hence, the study of the temporal fluctuation of the scattered light intensity should yield information about the dynamics of the scatterers. This is achieved by determining the temporal correlation of the number of photons arriving at a suitable detector<sup>78</sup>. The motion of the particles will also result in a Doppler shift between the incident and scattered light<sup>79</sup>. The remainder of this chapter is concerned with describing both the correlation and Doppler techniques. The latter will be extended to develop the theory behind the phase analysis technique. All the methods described can be broadly classified as dynamic light scattering.



### 3.3 PHOTON CORRELATION SPECTROSCOPY<sup>80, 81 82</sup>

Photon correlation spectroscopy (PCS) is also known as quasi-elastic light scattering (QELS) and intensity fluctuation spectroscopy (IFS).

Figure 3.1 is a schematic representation of the basic features of a PCS experiment. The difference between the magnitude of the incident wave vector,  $|\underline{k}_i|$ , and that of the scattered wave vector,  $|\underline{k}_s|$  is assumed to be negligible, i.e. the light is scattered quasielastically. A scattering vector,  $\underline{q}$ , is defined as

$$\underline{q} = \underline{k}_i - \underline{k}_s \quad (3.3)$$

and its magnitude

$$q = |\underline{q}| = \frac{4\pi}{\lambda} \sin \theta/2 \quad (3.4)$$

where  $\lambda$  is the wavelength of the light in the dispersion medium and  $\theta$  the scattering angle. For  $N$  identical, spherical particles undergoing translational motion in a fluid medium, the instantaneous value of the complex amplitude of the electric field of the light arriving at the detector is

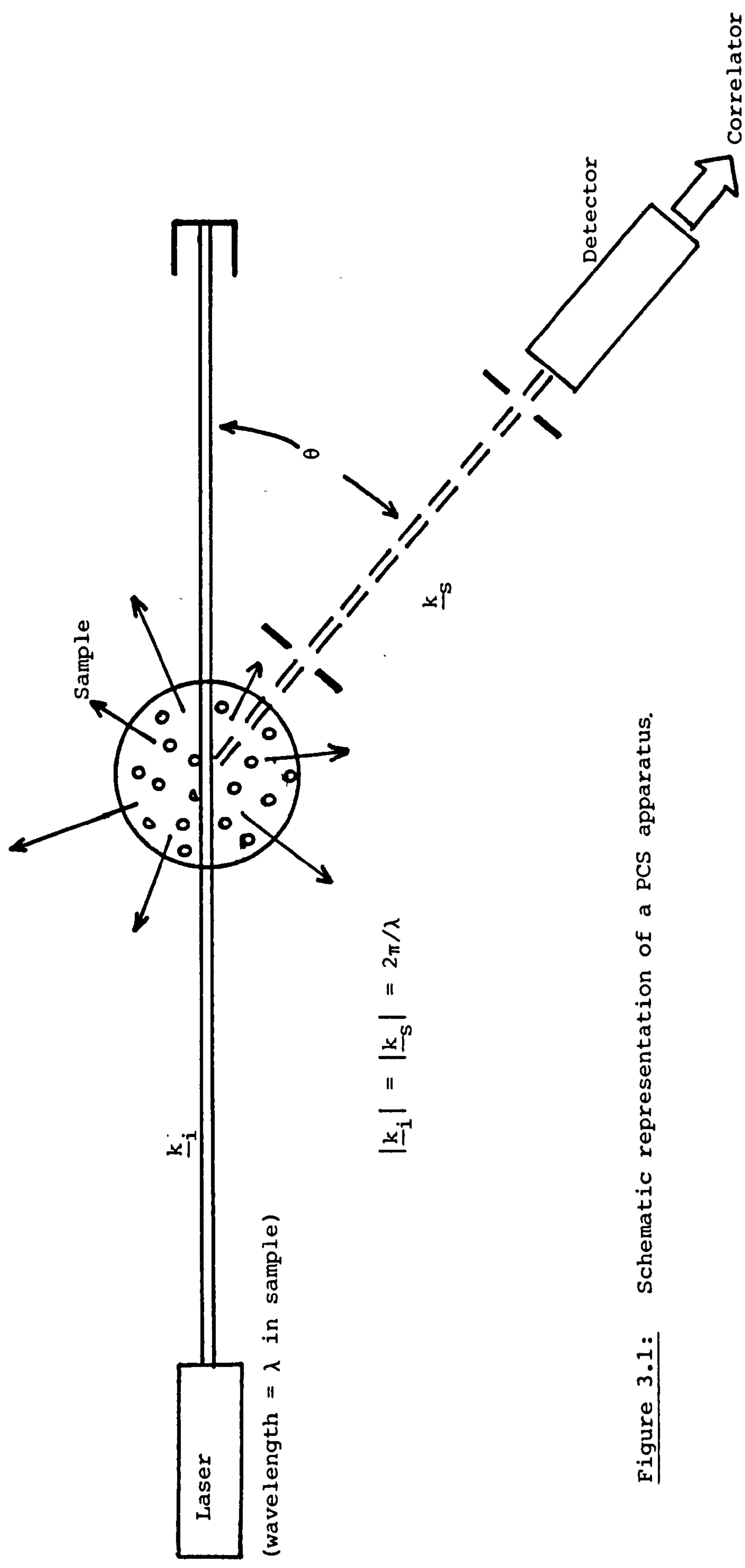


Figure 3.1: Schematic representation of a PCS apparatus.

$$E(\underline{q}, t) \propto \sum_{i=1}^N \exp(i \underline{q} \cdot \underline{r}_i(t)) \quad (3.5)$$

where  $\underline{r}_i(t)$  is the position of particle  $i$  at time  $t$ . Eq. (3.5) describes the speckle pattern obtained from the sample. The proportionality factor for this equation will reflect the spacial illumination profile, which will be dominated by the intensity cross section of the incident laser source, and the scattering properties of the sample. Generally, the proportionality factor can be assumed to be constant for a given experiment. As mentioned above, the random Brownian motion of the particle results in a random fluctuation of eq. (3.5). This random nature can be characterised by its first order autocorrelation function,  $G^{(1)}(\underline{q}, \tau)$ , given by

$$G^{(1)}(\underline{q}, \tau) = \langle E(\underline{q}, 0) E^*(\underline{q}, \tau) \rangle. \quad (3.6)$$

The normalised autocorrelation function,  $g^{(1)}(\underline{q}, \tau)$ , is given by

$$g^{(1)}(\underline{q}, \tau) = \frac{\langle E(\underline{q}, 0) E^*(\underline{q}, \tau) \rangle}{\langle |E(\underline{q}, 0)|^2 \rangle}. \quad (3.7)$$

The brackets ( $\langle \dots \rangle$ ) indicate ensemble averages. Hence,

$$g^{(1)}(\underline{q}, \tau) = \frac{\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^N \langle \exp(i \underline{q} \cdot [\underline{r}_i(0) - \underline{r}_j(\tau)]) \rangle}{\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^N \langle \exp(i \underline{q} \cdot [\underline{r}_i(0) - \underline{r}_j(0)]) \rangle} \quad (3.8)$$

The denominator term in eq. (3.8) can be thought of as a structure factor for the sample and the numerator as the dynamic structure factor. It is the latter that will contain translational information about the

sample. For dilute samples, interparticle interactions may be ignored. Thus, since each particle behaves independently, no correlation is obtained between different particles' positions and velocities; i.e. all the cross terms in eq. (3.8) disappear, leaving

$$g^{(1)}(q, \tau) = \langle \exp (i \underline{q} \cdot [\underline{r}_i(0) - \underline{r}_i(\tau)]) \rangle \quad (3.9)$$

If  $x(t)$  is the component of  $\underline{r}_i(t)$  in the direction of  $\underline{q}$ , then

$$g^{(1)}(q, \tau) = \langle \exp (i \underline{q} \cdot [x(0) - x(\tau)]) \rangle \quad (3.10)$$

$$= \langle \exp (-i \underline{q} \cdot \Delta x(\tau)) \rangle . \quad (3.11)$$

If  $V(t)$  is the velocity component of particles in the  $\underline{q}$ -direction at time  $t$ , then

$$\Delta x(\tau) = \int_0^{\tau} V(t) dt . \quad (3.12)$$

If it is assumed that the concentration fluctuation  $\delta c(x, t)$  follows the diffusion equation in one dimension<sup>7</sup>,

$$\frac{\partial (\delta c(x, t))}{\partial t} = D \frac{\partial^2 (\delta c(x, t))}{\partial x^2} , \quad (3.13)$$

then the following solution is obtained for eq. (3.13)

$$\delta c(x, t) = \frac{A}{(4Dt)^{\frac{1}{2}}} \exp (-x^2/4Dt) , \quad (3.14)$$

i.e. a Gaussian relation with Variance

$$\langle \Delta x(t)^2 \rangle = 2Dt, \quad (3.15)$$

where  $D$  is the translational diffusion coefficient. The Fourier transform of the concentration fluctuation given by Eq. (3.14) into reciprocal space is

$$\delta c(q, t) = \frac{A}{(4Dt)^{\frac{1}{2}}} \int_{-\infty}^{\infty} \exp(-x^2/4Dt) \exp(iqx) dx. \quad (3.16)$$

This is the standard integral of the form

$$\int_{-\infty}^{\infty} \exp(-p^2 x^2 \pm q'x) dx = \exp(q'^2/4p^2) \sqrt{\pi}/p, \quad (3.17)$$

with  $p^2 = 1/4Dt$  and  $q' = iq$ .

Hence,

$$\delta c(q, t) = A \sqrt{\pi} \exp(-q^2 Dt). \quad (3.18)$$

Initially,  $t=0$ ,  $\delta c(q, t) = \delta c(q, 0)$ , therefore,

$$\delta c(q, t) = \delta c(q, 0) \exp(-q^2 Dt), \quad (3.19)$$

which simply defines the correlation given in eq. (3.11), since

$\delta c(q, \tau) \propto G^{(1)}(q, \tau)$ . Hence, the average correlation function is

$$g^{(1)}(q, \tau) = \exp(-q^2 D\tau). \quad (3.20)$$

Therefore, the diffusion coefficient for a sample can be obtained readily from  $g^{(1)}(q, \tau)$ . For spherical particles,

$$D = kT/(6 \pi \eta a) , \quad (3.21)$$

where  $k$  is Boltzmann's constant,  $T$  the absolute temperature,  $\eta$  the bulk viscosity and  $a$  the particle radius.

Construction of the first order autocorrelation function experimentally requires that the amplitude of the scattered light be measured. In practice, however, photodetectors respond to the intensity of the photons falling on them, which is proportional to the square of the electric field amplitude. Hence, it is the intensity or second order autocorrelation function,  $G^{(2)}(q, \tau)$  that is obtained. This is also known as the photon correlation function,

$$G^{(2)}(q, \tau) = \langle n(0) n(\tau) \rangle , \quad (3.22)$$

where  $n(t)$  represents the output from the detector at a time  $t$ .

$n(t)$  depends on the intensity of the scattered light,  $I(t) (=E(t)E^*(t))$ .

The response of the detector should be such that the expectation of

$n(t)$  must be linear with that for  $I(t)$ , and that the statistics of

$I(t)$  must completely determine those of  $n(t)$ <sup>82</sup>. With these requirements,

$$G^{(2)}(q, \tau) = \langle E(0)E^*(0)E(\tau)E^*(\tau) \rangle , \quad (3.23)$$

and, for the normalised function,

$$g^{(2)}(q, \tau) = \frac{\langle E(0)E^*(0)E(\tau)E^*(\tau) \rangle}{\langle E(0)E^*(0) \rangle} . \quad (3.24)$$

If the optical field obeys Gaussian statistics,  $g^{(2)}(q, \tau)$  and  $g^{(1)}(q, \tau)$  are related by<sup>82, 83</sup>

$$g^{(2)}(q, \tau) = 1 + |g^{(1)}(q, \tau)|^2, \quad (3.25)$$

known as the Siegert relation<sup>84</sup>. Hence,

$$g^{(2)}(q, \tau) = 1 + \exp(-2Dq^2\tau). \quad (3.26)$$

The determination of the intensity autocorrelation function requires that the coherence of the scattered light is maintained at the detector. In practice, this requires that a small field of view be used. Slight loss in coherence due to photons arriving at different parts of the detector and thermal shot noise from the detector result in eq. (3.26) becoming

$$g^{(2)}(q, \tau) = K + \beta \exp(-2Dq^2\tau). \quad (3.27)$$

$K$  represents a background component or baseline, which can be determined for  $\tau \gg 1/Dq^2$ .  $\beta$  can be calculated as  $g^{(2)}(q, 0) - K$ .

Calculation of the decay constant for the exponential in eq. (3.27), and, hence, determination of the diffusion coefficient for the particles in the sample, is straightforward. If the baseline value,  $K$ , is not known, Swinbourne's method<sup>85</sup> may be employed to obtain  $K$ ,  $\beta$  and  $Dq^2$ .



The expressions for  $g^{(1)}(q, \tau)$  and  $g^{(2)}(q, \tau)$  are valid for a sample consisting of identical, isotropic particles. In practice, however, a sample may be somewhat polydisperse. The main effect of this on the correlation function is that the decay constant for the exponential decreases with increasing  $\tau$ , since the diffusion coefficient decreases with increasing particle size. Hence, a plot of  $\ln |g^{(1)}(q, \tau)|$  against  $\tau$  will not yield the expected straight line of gradient  $-Dq^2$ , but a curve, with initial gradient  $-Dq^2$ . Pusey et al<sup>86</sup> have expanded  $\exp(-\Gamma\tau)$  around  $\exp(-\bar{\Gamma}\tau)$ , i.e.

$$\begin{aligned} \exp(-\Gamma\tau) &= \exp(-\bar{\Gamma}\tau) \exp[-(\Gamma-\bar{\Gamma})\tau] \\ &= \exp(-\bar{\Gamma}\tau) \times \left( 1 - (\Gamma-\bar{\Gamma})\tau + \frac{(\Gamma-\bar{\Gamma})^2 \tau^2}{2!} \right. \\ &\quad \left. - \frac{(\Gamma-\bar{\Gamma})^3 \tau^3}{3!} + \dots \right) \end{aligned} \quad (3.28)$$

The mean decay rate,  $\bar{\Gamma}$ , is defined by

$$\bar{\Gamma} = \int_0^\infty \Gamma G(\Gamma) d\Gamma, \quad (3.29)$$

where  $G(\Gamma)$  is the normalised distribution function of the decay rates.

Thus,

$$|g^{(1)}(q, \tau)| = \exp(-\bar{\Gamma}\tau) \left\{ 1 + \frac{\mu_2 \tau^2}{2!} - \frac{\mu_3 \tau^3}{3!} + \dots \right\} \quad (3.30)$$

where  $\mu_i$  is the  $i^{\text{th}}$  moment about the mean of  $G(\Gamma)$ , i.e.

$$\mu_i = \int_0^{\infty} (\Gamma - \bar{\Gamma})^i G(\Gamma) d\Gamma . \quad (3.31)$$

Using the identity  $\ln(1 + x) = x - \frac{1}{2} x^2 + \dots$  ,

$$\begin{aligned} \ln |g^{(1)}(q, \tau)| &= -\bar{\Gamma}\tau + \frac{1}{2!} \frac{\mu_2}{\bar{\Gamma}^2} (\bar{\Gamma}\tau)^2 \\ &\quad - \frac{1}{3!} \frac{\mu_3}{\bar{\Gamma}^3} (\bar{\Gamma}\tau)^3 + \dots . \end{aligned} \quad (3.32)$$

For a monodisperse system (i.e. a single exponential decay in the correlation function),  $\mu_i = 0$ . A plot of  $d \ln |g^{(1)}(q, \tau)| / d\tau$  vs.  $\tau$  should yield a function with a linear gradient term (i.e. dependent on  $\tau$ ),  $\mu_2$ . From this, an estimation of the z-average normalised variance of the D distribution ( $\bar{\Gamma} = Dq^2$ ) can be estimated,

$$\mu_2 / \bar{\Gamma}^2 = \left| (\bar{D}^2)_z - \bar{D}_z^2 \right| / \bar{D}_z^2 . \quad (3.33)$$

Photon correlation spectroscopy is now a widely used tool for characterising colloidal dispersions and for studying the diffusion of polymers in solution.

### 3.4 LASER-DOPPLER METHODS

#### 3.4.1 RELATING POWER SPECTRA TO CORRELATION FUNCTIONS

It was briefly mentioned in §3.2 that the motion of the scattering particles results in a Doppler shift between the incident and scattered light. Clearly, scattering from a sample containing randomly diffusing particles will result in a broadening of the frequency distribution of the scattered light. A quantitative description of this broadening is required. If  $\omega_o$  and  $\omega_s$  are the incident and scattered light radial frequencies respectively, then the Doppler shift,  $\omega$ , is simply  $\omega_s - \omega_o$ . Let  $I(\omega)$  be the power spectrum of the scattered light, centred around  $\omega_o$ . According to the Wiener-Khintchine theorem<sup>87</sup>, if some event  $x(t)$  is generated by a stationary random process, the ensemble power spectral density,  $\phi(\omega)$ , is defined by

$$\phi(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} C(\tau) \exp(-i\omega\tau) d\tau, \quad (3.34)$$

where  $C(\tau)$  is the ensemble correlation function. Thus, for the scattered light power spectrum,

$$I(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} g^{(1)}(q, \tau) \exp(-i\omega\tau) d\tau, \quad (3.35)$$

i.e. Fourier transformation of the correlation function from the time domain to the frequency domain will yield the power spectrum, or

Doppler spectrum. For identical, spherical particles,

$$I(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \exp(-Dq^2\tau) \exp(-i\omega\tau) d\tau. \quad (3.36)$$

The Fourier transform of an exponential such as in eq. (3.36) is well-known<sup>88</sup>. In this case,

$$I(\omega) = 2Dq^2 / (\omega^2 + (Dq^2)^2), \quad (3.37)$$

i.e. the power spectrum for the scattered light from identical, randomly diffusing particles is a Lorentzian, centered around the incident frequency, and with a width at half height of  $2Dq^2$  rad. In principle, the power spectrum should also be obtainable by Fourier transformation of the scattered light signal from the time domain into the frequency domain. With the "homodyne" method so far described, this is not directly possible, since it would require a spectrum analyser capable of Fourier transforming signals with  $\omega_0 \sim 6 \times 10^{14} \text{ rad s}^{-1}$ , and yielding spectra with a resolution of  $10^0$  to  $10^2 \text{ Hz}$  ( $2\pi$  to  $200\pi \text{ rad s}^{-1}$ ). The "heterodyne" method, described below, does allow frequency spectra to be obtained for the scattered light.

#### 3.4.2 OBTAINING DOPPLER SPECTRA BY HETERODYNING (LIGHT BEATING)

A commonly used method in radio electronics for determining small frequency shifts around a large "carrier" or "centre" frequency

is that of heterodyning. For example, frequency modulated radio broadcasts are demodulated by mixing the received signal with a signal whose frequency is the centre frequency of the transmitted signal. The two signals add together to produce a "beat" frequency, equal to the frequency difference. Hence, by analogy, if the scattered light signal (i.e. the photo current from a detector) were to be mixed with a signal whose frequency is that of the incident laser source, a beat frequency equal to the Doppler frequency would be obtained. This is not possible, however, since the frequency response of typical detectors diminishes rapidly above a few MHz - GHz. Fortunately, though, the non-linear response of the photo detectors usually used in dynamic light scattering experiments can be exploited. Forrester et al<sup>89</sup> discovered that when the Zeeman-split emission line from mercury (546.1 nm) fell on a square law response detector, the two strong optical components beat with each other to produce a photocurrent with a strong frequency component at  $\sim 10^{10}$  Hz. Thus, if one of the light waves arriving at the detector is that from the laser before it is scattered, and the other is the scattered radiation, then the two will heterodyne to produce a photo current from the detector with a frequency equal to the Doppler shift. This photocurrent can then be fed into a spectrum analyser, and the power spectrum obtained, with 0Hz representing  $\omega_0$ . Note, however, that only the magnitude of the Doppler shift is obtained.

### 3.4.3 INFLUENCE OF A CONSTANT, COLLECTIVE VELOCITY

The above considerations for autocorrelation functions and Doppler spectra have assumed that the average velocity of the particles in the sample is zero. It is now necessary to consider the effect of a constant, collective motion in addition to the random diffusive motion. Such a constant motion may be due to electrophoresis in a uniform d.c. electric field. Let  $V_c$  be the constant velocity motion in the direction of  $\underline{q}$ . Then, the diffusion equation in eq. (3.13) becomes

$$\frac{\partial (\delta_c(x,t))}{\partial t} = D \frac{\partial^2 (\delta_c(x,t))}{\partial x^2} + V_c \frac{\partial (\delta_c(x,t))}{\partial x} . \quad (3.38)$$

The first order autocorrelation function obtained for eq. (3.38) is<sup>90</sup>

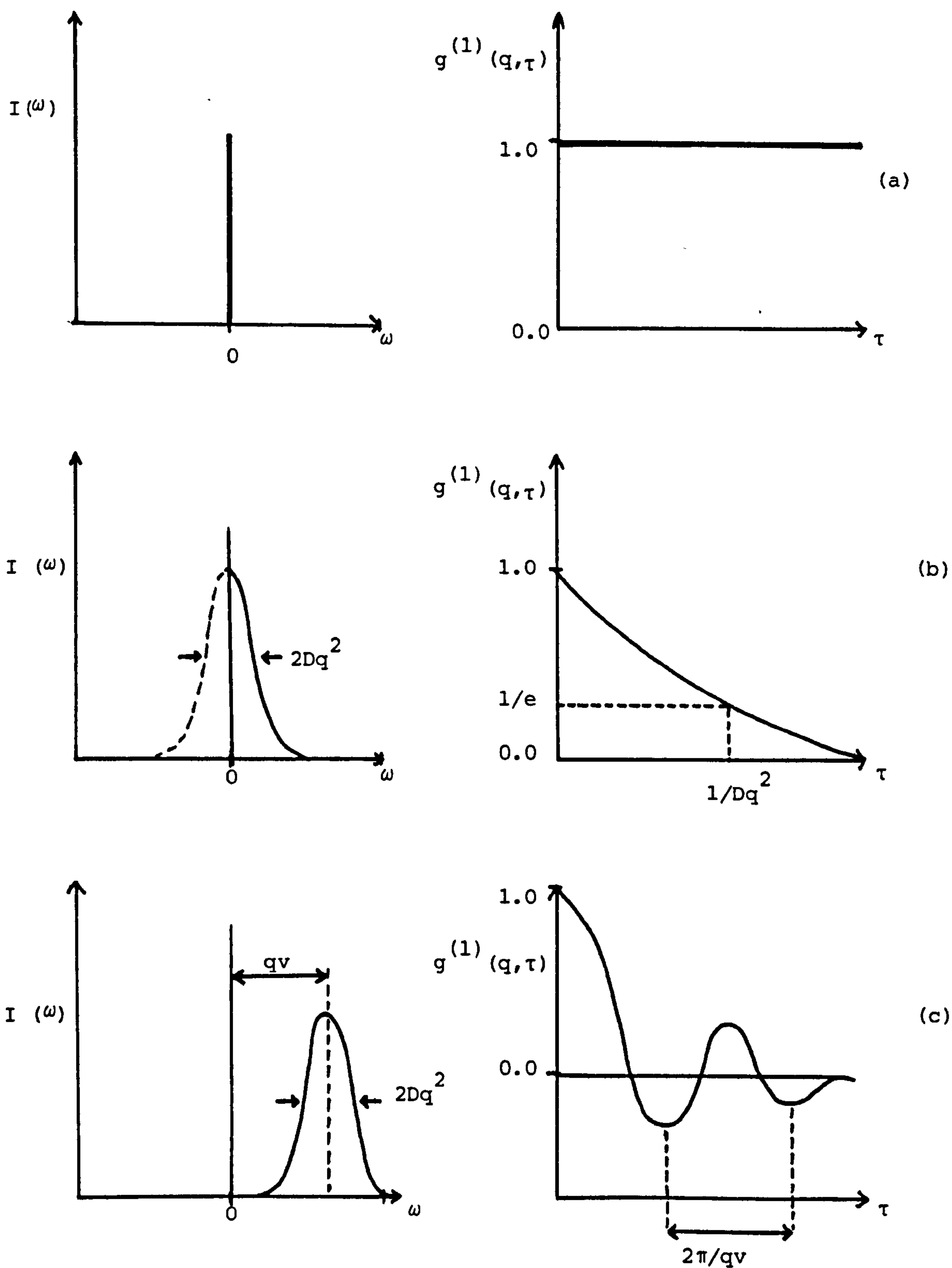
$$g^{(1)}(q,\tau) = \exp(-iqV_c\tau) \exp(-q^2 D\tau) , \quad (3.39)$$

and the corresponding power spectrum is

$$I(\omega) = 2Dq^2 / ((\omega + qV_c)^2 + (Dq^2)^2) , \quad (3.40)$$

i.e. in the presence of a constant velocity,  $V_c$ , in the direction of  $\underline{q}$ , the power spectrum described by eq. (3.37) is simply shifted by a frequency  $qV_c$ . Figure 3.2 illustrates the frequency spectra and correlation functions for the incident light source, the scattered light from a diffusing sample and for the scattered light from a diffusing sample with a constant velocity term. If the

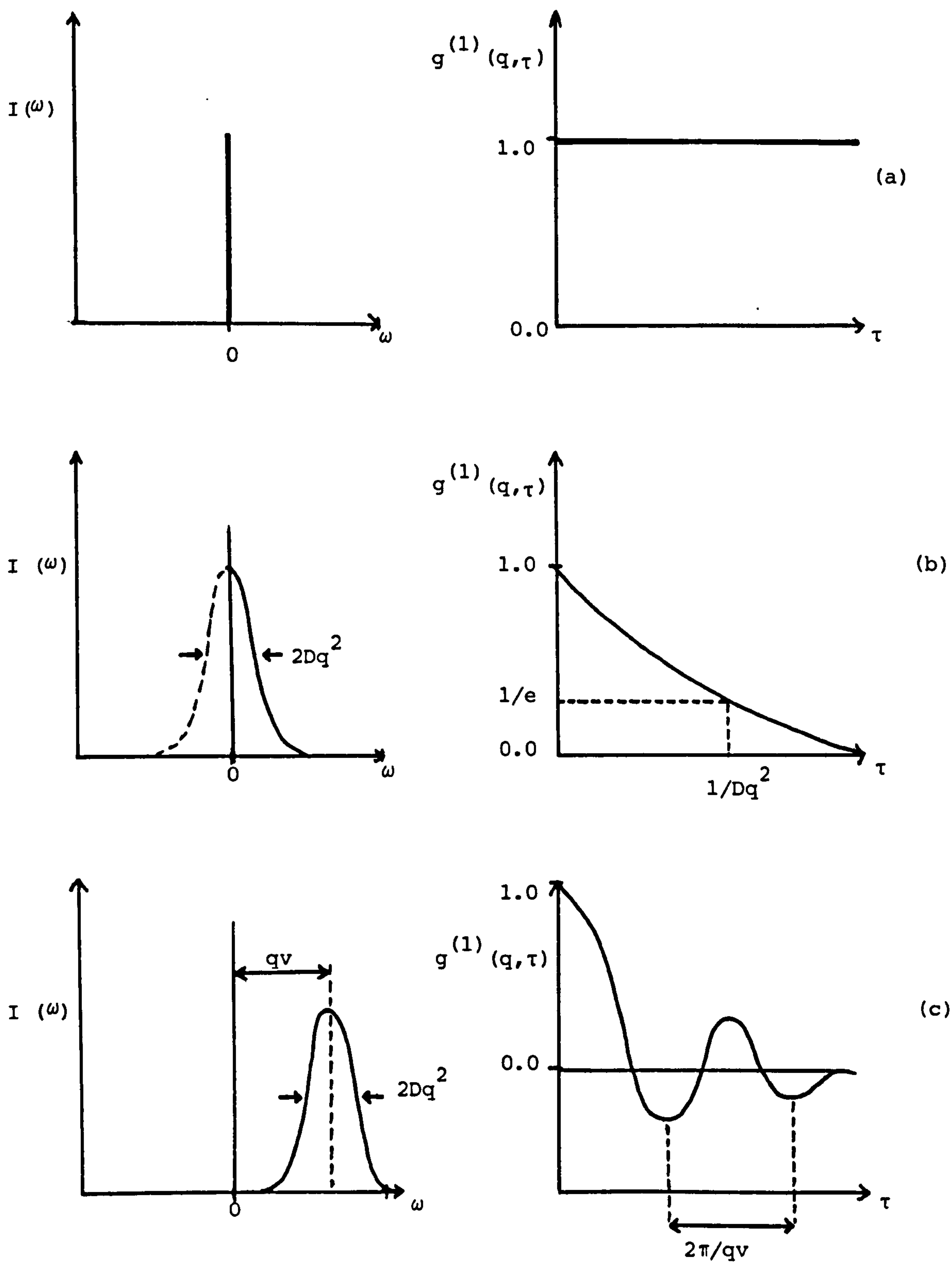




**Figure 3.2:** Normalised first order correlation functions and power spectra for :

- (a) incident laser radiation source
- (b) scattered light from sample exhibiting diffusion only
- (c) scattered light from sample exhibiting diffusion and a constant, collective velocity,  $v$ .





**Figure 3.2:** Normalised first order correlation functions and power spectra for :

- (a) incident laser radiation source
- (b) scattered light from sample exhibiting diffusion only
- (c) scattered light from sample exhibiting diffusion and a constant, collective velocity,  $v$ .

constant velocity arises through the motion of particles in the sample due to an applied electric field, then it is possible to determine the electrophoretic mobility of the sample<sup>90</sup>. Figure 3.3 shows an experimental arrangement for the laser-Doppler electrophoresis (LDE) method. The particles migrate in a direction perpendicular to the electrodes, which are parallel to the incident beam. Hence, the velocity component in the correlation function/power spectrum is

$$V_c = V_E \cos (\theta/2) = \mu E \cos (\theta/2) . \quad (3.41)$$

Thus, eq. (3.39) becomes

$$g^{(1)}(q, \tau) = \exp (-iq\mu E \cos (\theta/2)) \exp (-q^2 D \tau) \quad (3.42)$$

and eq. (3.40) becomes

$$I(\omega) = 2Dq^2 / ((\omega + q\mu E \cos (\theta/2))^2 + (Dq^2)^2) . \quad (3.43)$$

Thus, in principle, both the electrophoretic mobility and diffusion coefficient of a sample can be determined simultaneously.

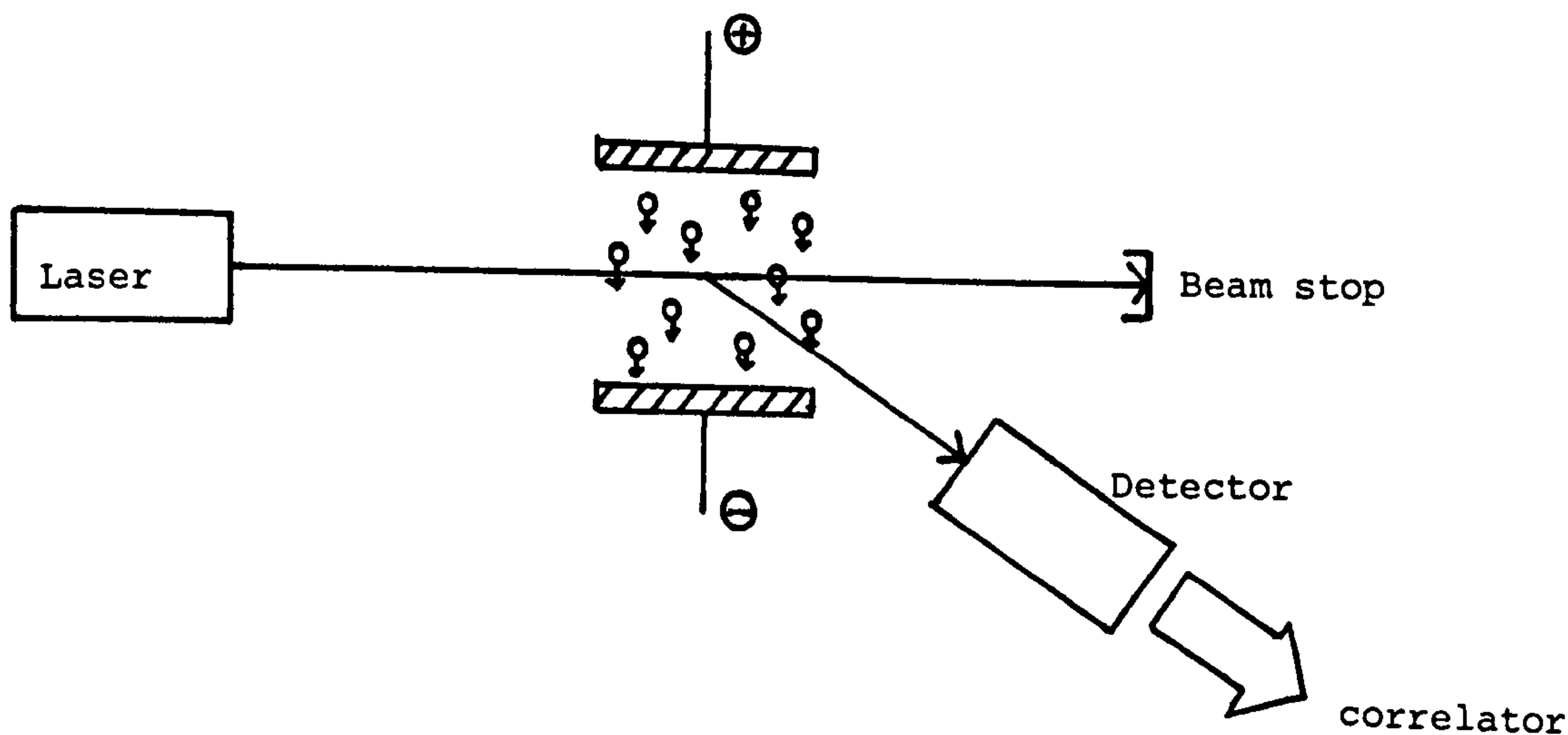


Figure 3.3: Schematic representation of Ware and Flygare's<sup>89</sup> laser-Doppler electrophoresis apparatus.

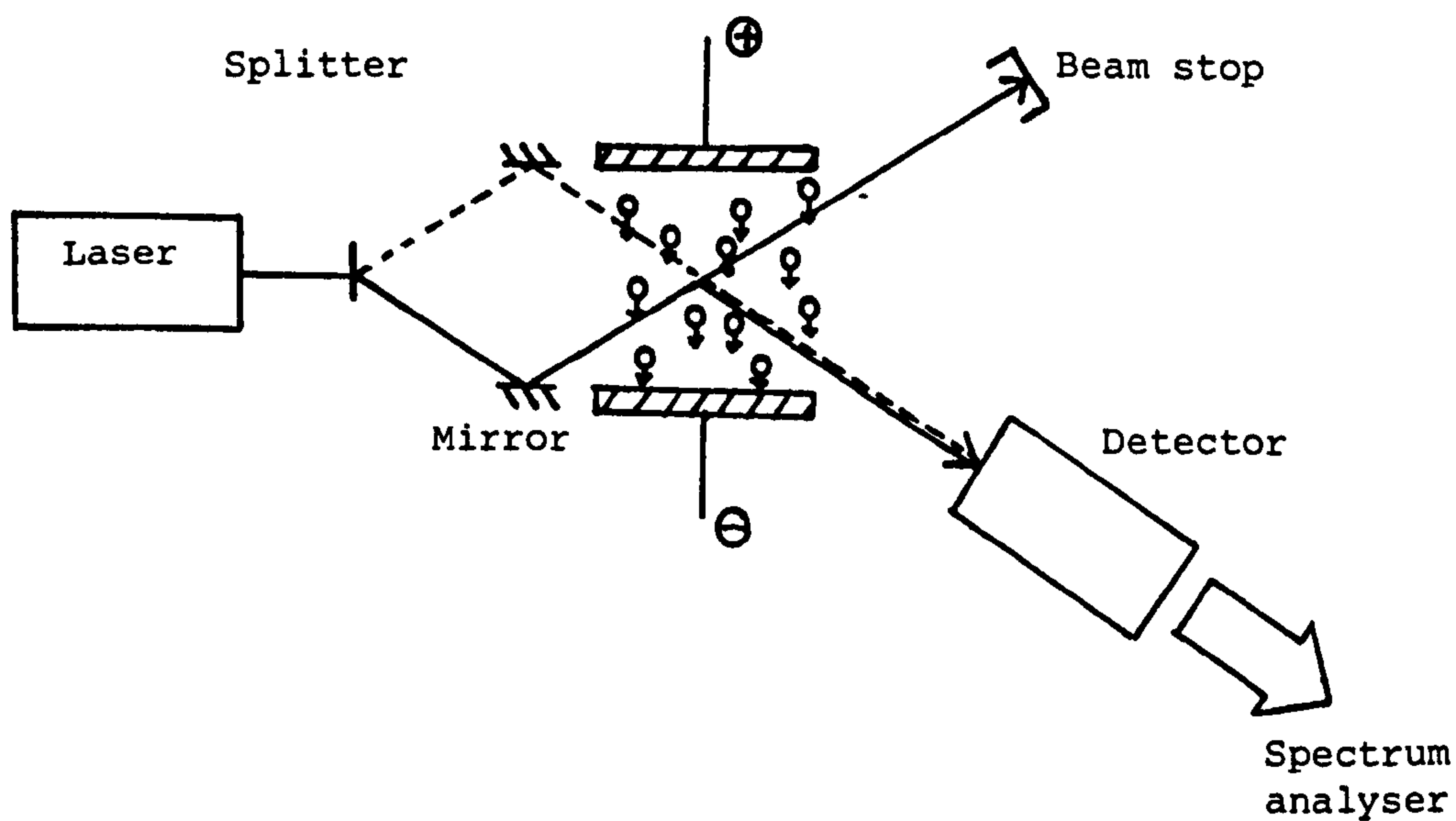


Figure 3.4: Reference beam heterodyne velocimeter used for electrophoresis studies by Yoshimura et al(94).

#### 3.4.4 APPLICATION TO SAMPLES WITH MORE THAN ONE SPECIES OF PARTICLE

The aspects of dynamic light scattering discussed so far apply to dispersions containing many identical particles. For a sample containing two species of particles, the correlation function becomes rather complicated. The desire to study the diffusion of multicomponent dispersions led Ware and Flygare to develop the method outlined above<sup>91</sup>. Assuming that the different species have different electrophoretic mobilities, it was proposed that the scattered light for each species would appear as distinct Lorentzian peaks in the scattered light power spectrum. Ware and Flygare demonstrated this with a study of bovine serumalbumin (BSA) in aqueous buffer<sup>92</sup>. The BSA exists as both monomers and dimers in solution, hence two distinct Lorentzian peaks are found in the power spectrum when the sample is subjected to an applied electric field. In general, if more than one component is present in the sample, then it is easier to interpret the power spectrum than the correlation function. It should be noted at this point that the above methodology is not capable of resolving the direction of the electrophoretic motion. This allows square wave electric fields to be used, thereby minimising polarisation effects due to macroscopic mass transport of the particles.

Ware and Flygare relied upon the sample cell to scatter some unshifted incident radiation to provide the light with which to mix

the scattered light. Yeh and Cummins<sup>93</sup> split the incident radiation source into two beams. One was sent directly to the detector, the other was passed through the sample under investigation and then to the detector. Yoshimura et al<sup>94</sup> used the heterodyne method to study the electrophoresis of polystyrene latex particles in aqueous buffer solutions. Figure 3.4 is a schematic representation of their experiment. This "reference beam" method, like the single beam, homodyne method, requires that the coherence of the scattered light be maintained, hence the usable area of the photodetector needs to be minimal. An alternative method, which removes the coherence restriction, is described below.

#### 3.4.5 DUAL BEAM / REAL FRINGE HETERODYNE VELOCIMETER

Consider the identical laser beams intersecting at an angle  $\theta$  in, for example, a colloidal dispersion, as represented in Figure 3.5. The scattered light arriving at the detector due to beam A will possess a Doppler shift  $\Delta\omega_A$  given by

$$\Delta\omega_A = (\underline{k}_{i,A} - \underline{k}_s) \cdot \underline{V} \quad , \quad (3.44)$$

where  $\underline{V}$  is the velocity component of the particle motion in the direction of  $\underline{q}$ . In a similar way, the scattered light due to beam B will possess a Doppler shift  $\Delta\omega_B$ ,

$$\Delta\omega_B = (\underline{k}_{i,B} - \underline{k}_s) \cdot \underline{V} \quad . \quad (3.45)$$

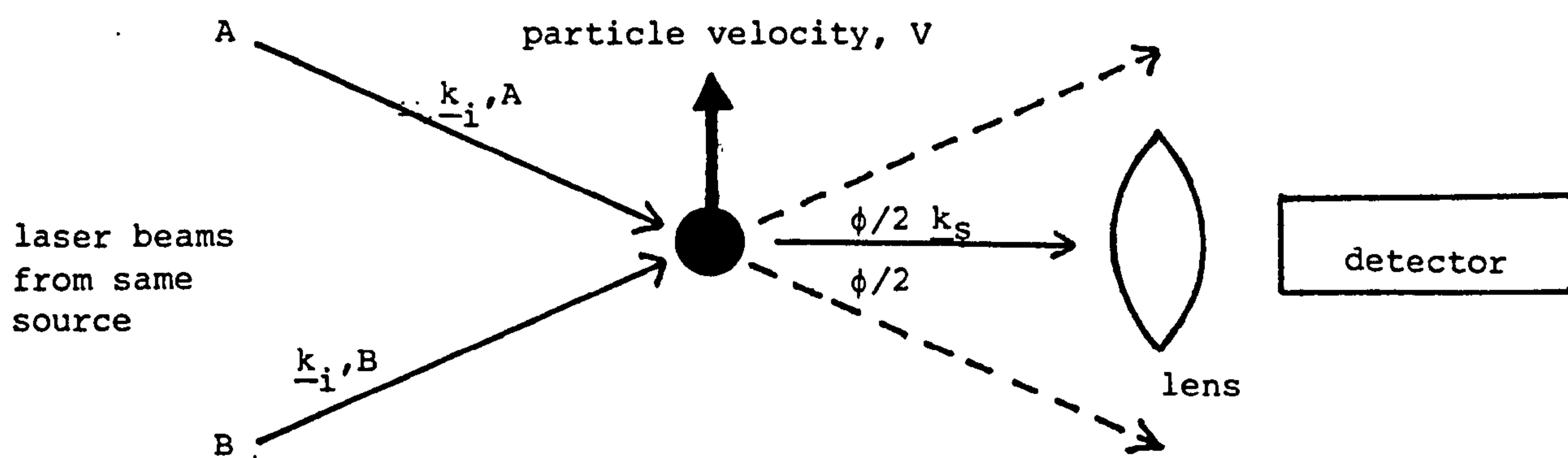


Figure 3.5: Schematic representation of a dual beam/real fringe heterodyne velocimeter.

Hence, the scattered light from both beams will heterodyne at the (non-linear) detector to produce a detector signal with a frequency

$$\Delta\omega_A - \Delta\omega_B = (\underline{k}_{i,A} - \underline{k}_{i,B}) \cdot \underline{v} = qV \quad (3.46)$$

$$= \frac{4\pi \sin(\theta/2)}{\lambda} V \text{ rad s}^{-1} \quad (3.47)$$

$$= \frac{2 \sin(\theta/2)}{\lambda} V \text{ Hz} , \quad (3.48)$$

which is directly analogous to the Doppler signal obtained for the "reference" beam method. Thus, the signal from the photodetector can be fed into a spectrum analyser to yield a scattered light power spectrum or Doppler spectrum. It is important to realise, however, that the detector signal frequency depends on the angle of intersection between the two incident beams and not the position of the detector in space. Thus, this arrangement does not require that the scattered light maintain its coherence at the detector. This allows wider fields of view to be employed, thereby enabling more of the scattered light to be used. This offers a clear advantage over the "reference" beam method, particularly for samples with weak scattering properties. Since the detector signal arises through the difference between the Doppler shifts of the two illuminating beams, this method is also called the "differential heterodyne" technique<sup>95</sup>.



### 3.4.6 FRINGE MODEL FOR DUAL BEAM DIFFERENTIAL HETERODYNE VELOCIMETER<sup>96</sup>

The intersection of the two laser beams in the manner described above will result in interference effects. Figure 3.6 illustrates how the constructive and destructive interference leads to the creation of interference fringes, normal to the direction of  $\underline{q}$  ( $\underline{k}_{i,A} - \underline{k}_{i,B}$ ). As can be seen geometrically, the fringe spacing,  $d$ , is given by

$$d = \lambda / (2 \sin (\theta/2)) \quad (3.49)$$

Thus, if a scattering body traverses the fringe volume in the  $\underline{q}$ -direction, the illumination of that body will vary sinusoidally as it crosses alternate light and dark fringes. The scattered light intensity, observed by the detector<sup>95</sup>, will vary sinusoidally with a radial frequency,  $\omega$ , given by

$$\omega = 2 \pi \cdot v/d \quad (3.50)$$

$$= \frac{4 \pi \sin (\theta/2)}{\lambda} \quad (3.51)$$

where  $v$  is the velocity component of the moving scatterer in the direction of  $\underline{q}$ . Clearly, from eq. (3.4),

$$\omega = v \cdot |\underline{q}| = v \cdot q \quad (3.52)$$

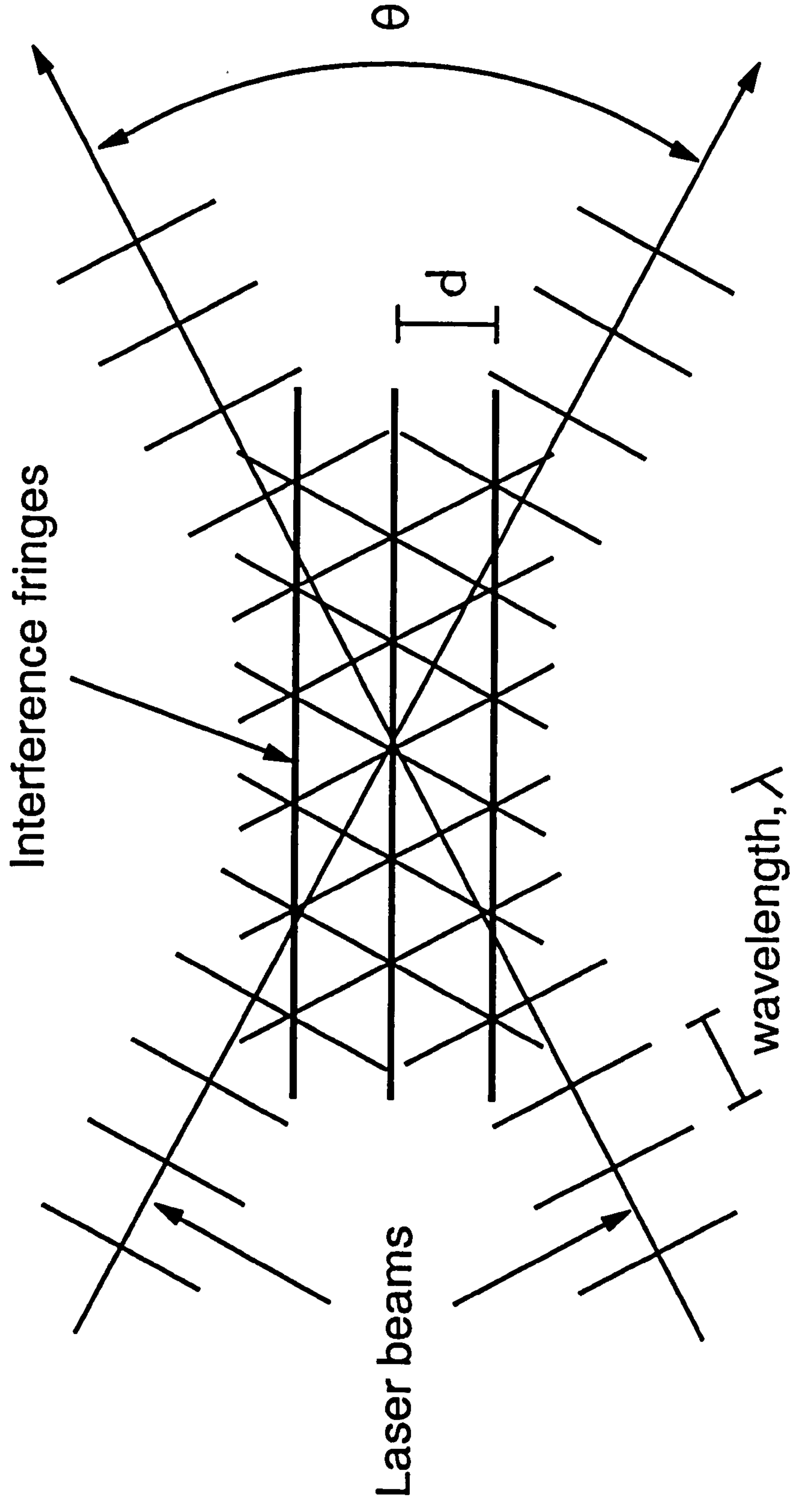


Figure 3.6: Interference fringes resulting from the intersection

of two laser beams.

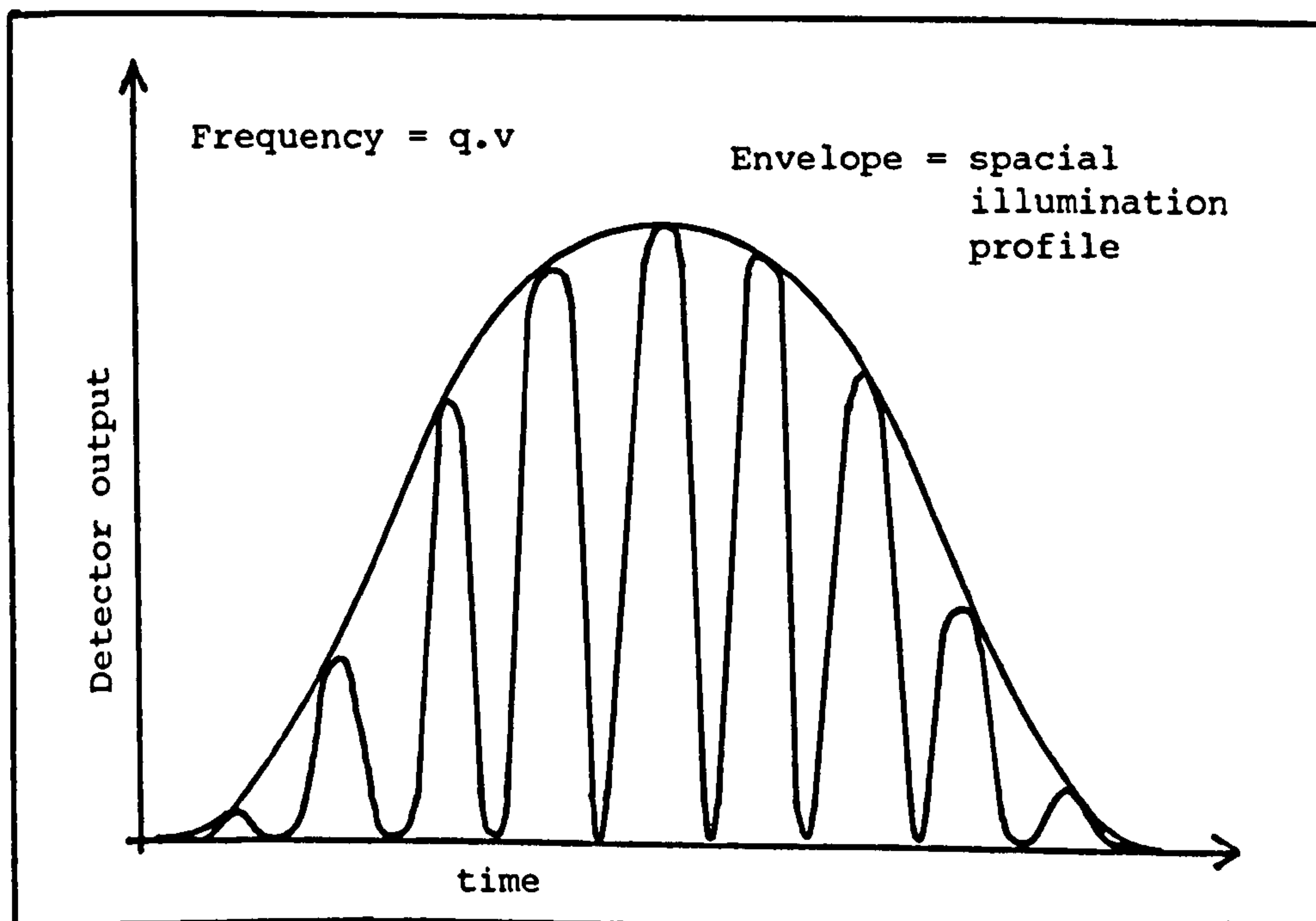
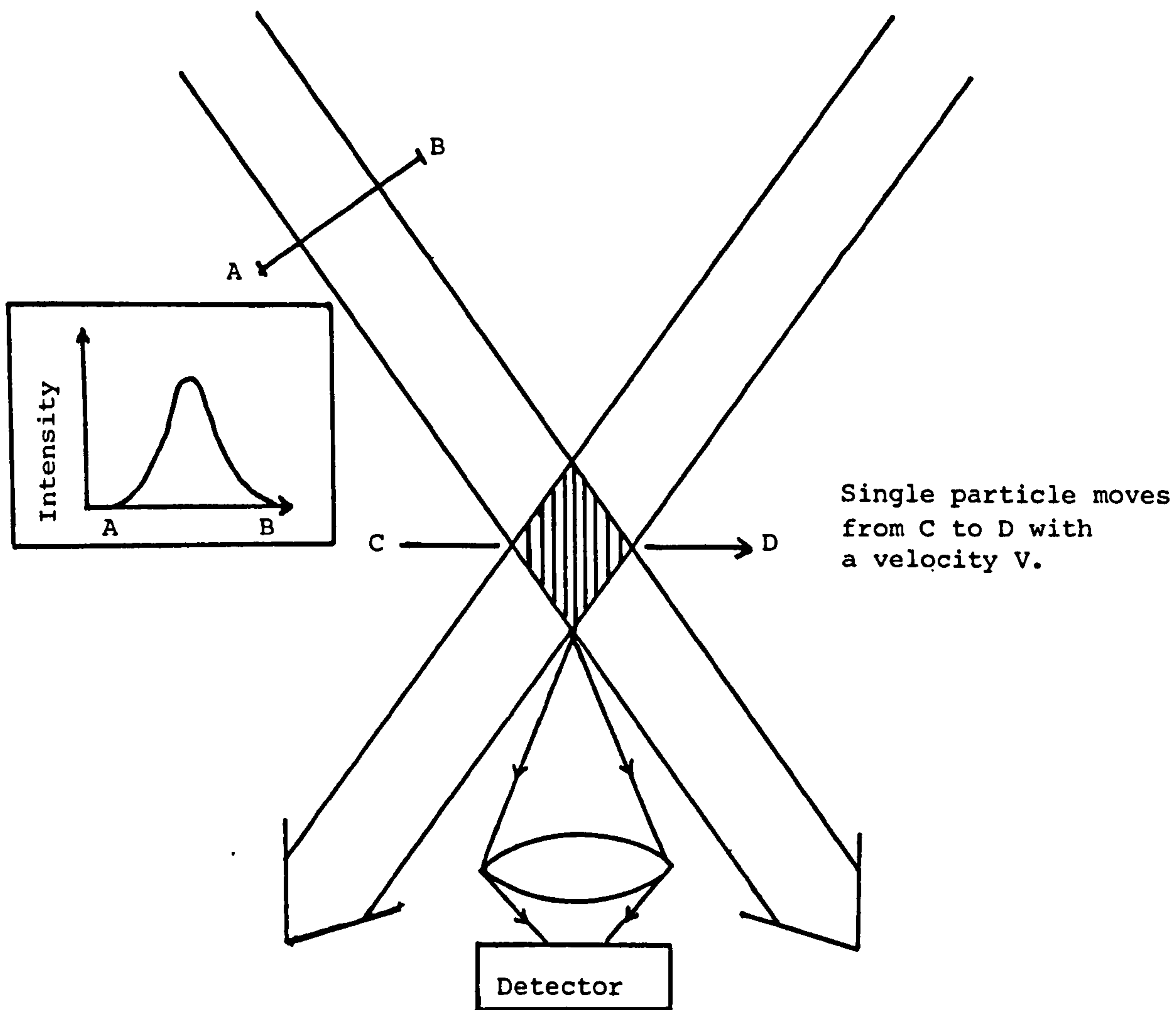
which is identical to eq. (3.46). Hence, the origin of the Doppler signal can be adequately explained by the concept of interference fringes. This is easier to visualise than the explanation given in § 3.4.5.

### 3.5 PROCESSING THE DOPPLER SIGNAL FROM A REAL FRINGE VELOCIMETER

In order to appreciate the practical limitations of the real fringe technique, a description of the general characteristics of the detector signal is useful.

#### 3.5.1 FRINGE VOLUME SPACIAL ILLUMINATION PROFILE

The intensity cross section of the laser source used for a laser-Doppler technique is usually Gaussian. Hence, the intensity of the fringes generated by the intersection of two such laser beams will follow a Gaussian distribution. Thus, if a single particle traverses the whole of the fringe region in the  $q$ -direction, the detector signal will consist of a high frequency component (the Doppler frequency) with a Gaussian envelope. This is illustrated in Figure 3.7. In practice, a typical fringe volume may be 0.5mm wide (in the  $q$ -direction) and will contain  $\sim 1000$  fringes. The optical arrangement is usually designed so that the detector only views a small, central part of the fringe volume, and, hence, the spacial illumination profile can be considered flat.



**Figure 3.7:** Origin of spacial illumination profile and Doppler signal for one particle traversing fringe volume.

### 3.5.2 PROCESSING SIGNALS DUE TO SINGLE PARTICLES

An ideal signal, such as depicted in Figure 3.7, is relatively straightforward to process. High pass filtering of the signal yields a signal such as shown in Figure 3.8. The determination of the Doppler period is easily achieved using frequency counting techniques<sup>95</sup>. In practice, the detector signal will contain high frequency noise, which is readily removed by low-pass filtering of the signal. (See inset in Figure 3.8).

### 3.5.3 SIGNALS DUE TO MANY PARTICLES

The scattered light detected from a dilute colloidal dispersion will usually result from the motion of many particles in the fringe volume at a given instant. The signals from each particle will generate random interference events. Figure 3.9 represents the detected signal for multiple particle scattering.

The processing of such signals using simple counting techniques is difficult. Determination of the frequency spectrum, however, is straightforward. A spectrum analyser can be employed to do this. Modern spectrum analysers employ fast Fourier transform processing and can be considered real time processors. Older analysers, employing frequency sweeping techniques, can present problems. These

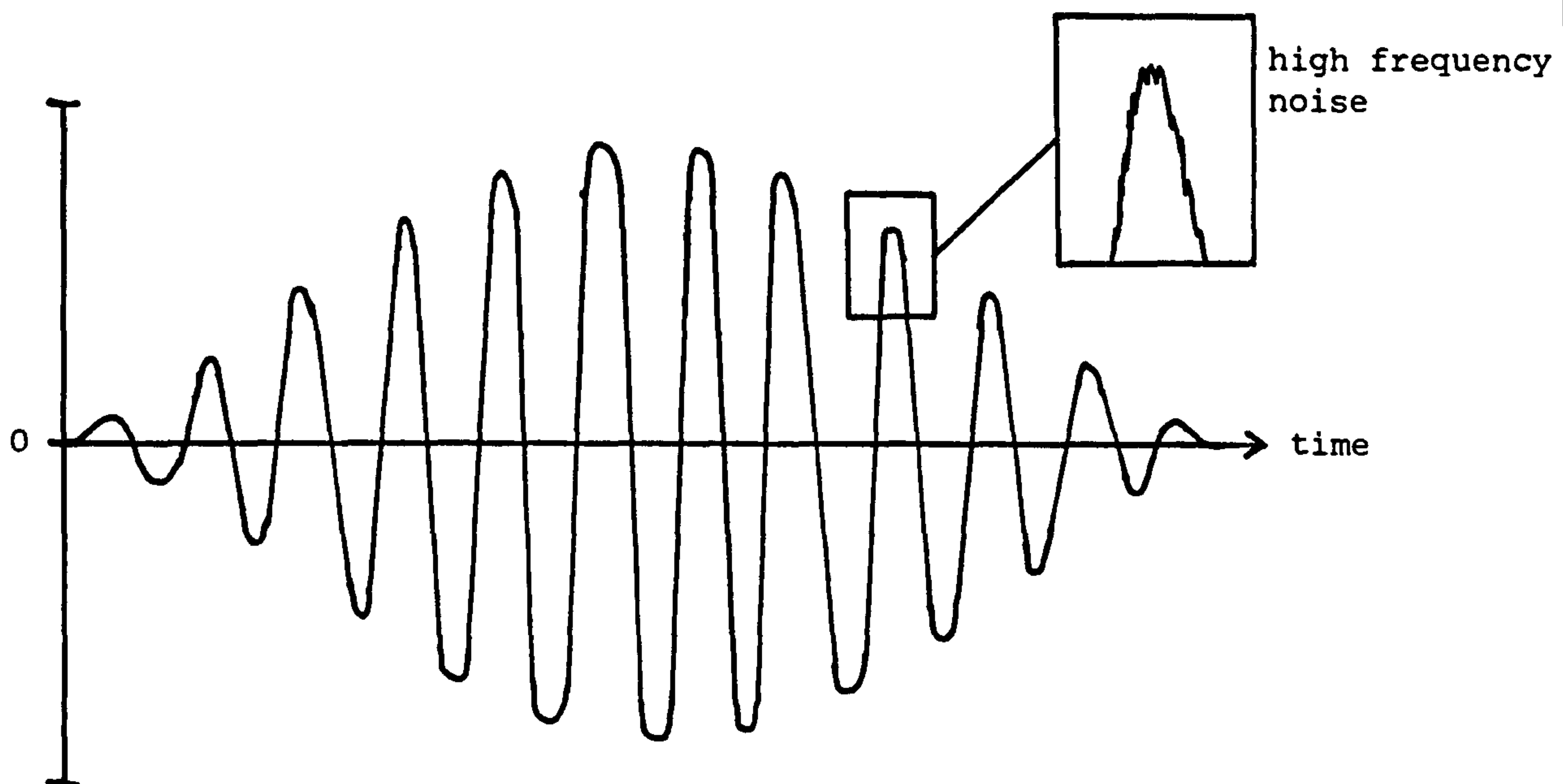


Figure 3.8: High-pass filtered Doppler signal such as shown in Figure 3.7.

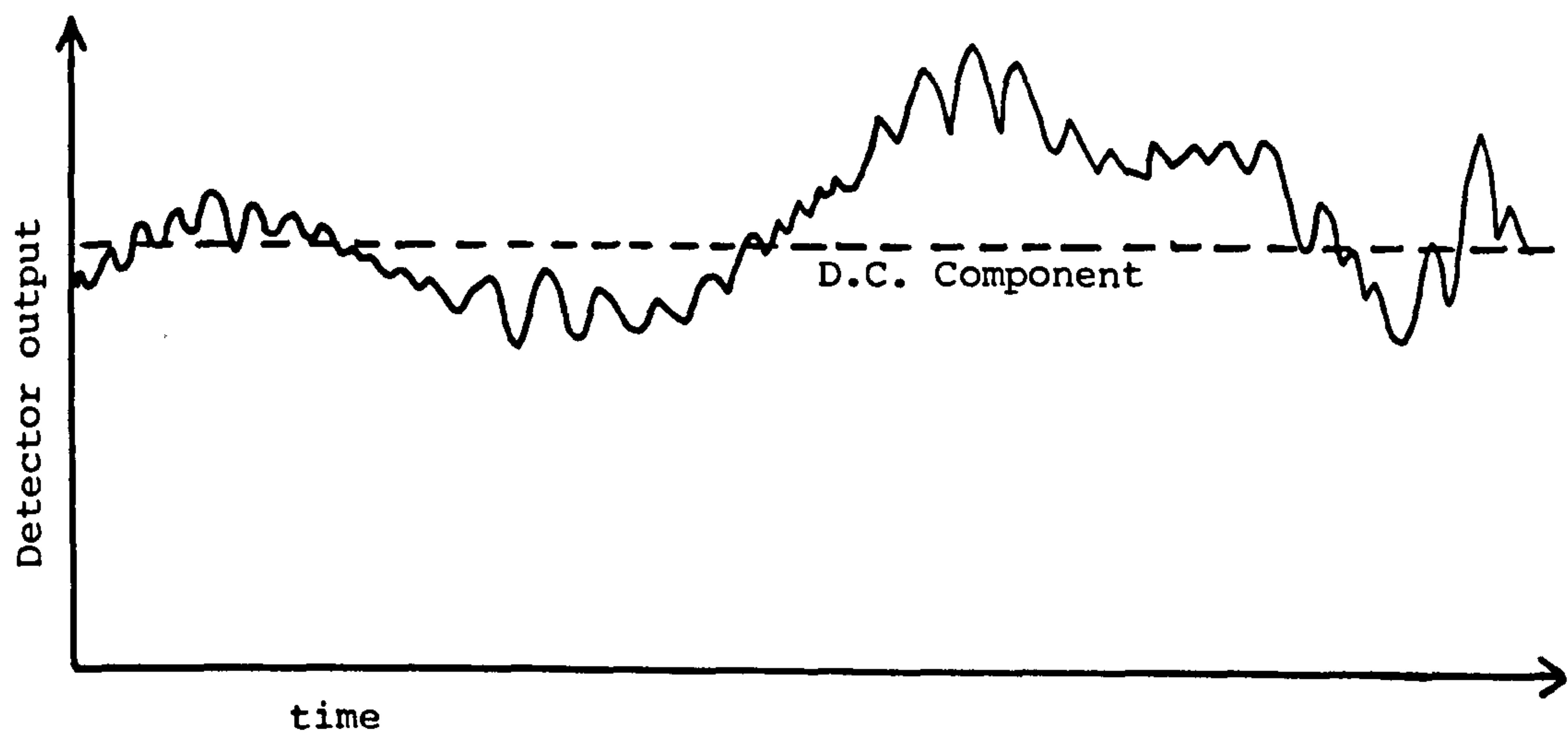


Figure 3.9: Doppler signal due to many particles moving in the fringe volume.



are discussed by Durst et al<sup>95</sup>. The choice of spectrum analyser is dictated by the magnitude of the velocities that are to be measured. The study of electrophoresis involves the measurement of electrophoretic velocities of the order of  $10 - 100 \mu\text{m s}^{-1}$ . Typical fringe spacings are of the order of  $0.5 - 2 \mu\text{m}$ . Thus, Doppler frequencies of the order of  $10 - 200 \text{ Hz}$  are expected. Audio frequency spectrum analysers ( $0$  to  $25 \text{ kHz}$ ) are generally available and are ideally suited to the study of laser-Doppler signals arising through electrophoresis. However, there is a problem. The signal depicted in Figure 3.9 contains a low frequency component as well as the higher frequency component of interest. The frequency spectrum for this signal will be of the form shown in Figure 3.10. The low frequency component in the signal gives rise to the "pedestal" shown in the frequency spectrum. It is necessary to consider two aspects of this signal. Firstly, what parameters influence the relative amplitude of the pedestal and the desired Doppler peak and, secondly, what factors determine the separation between the two peaks.

The ratio of the amplitudes of the desired Doppler peak and the low frequency pedestal is proportional to the signal to noise ratio, which depends on<sup>95</sup> :

- (i) laser power,
- (ii) ratio of particle size to fringe spacing,



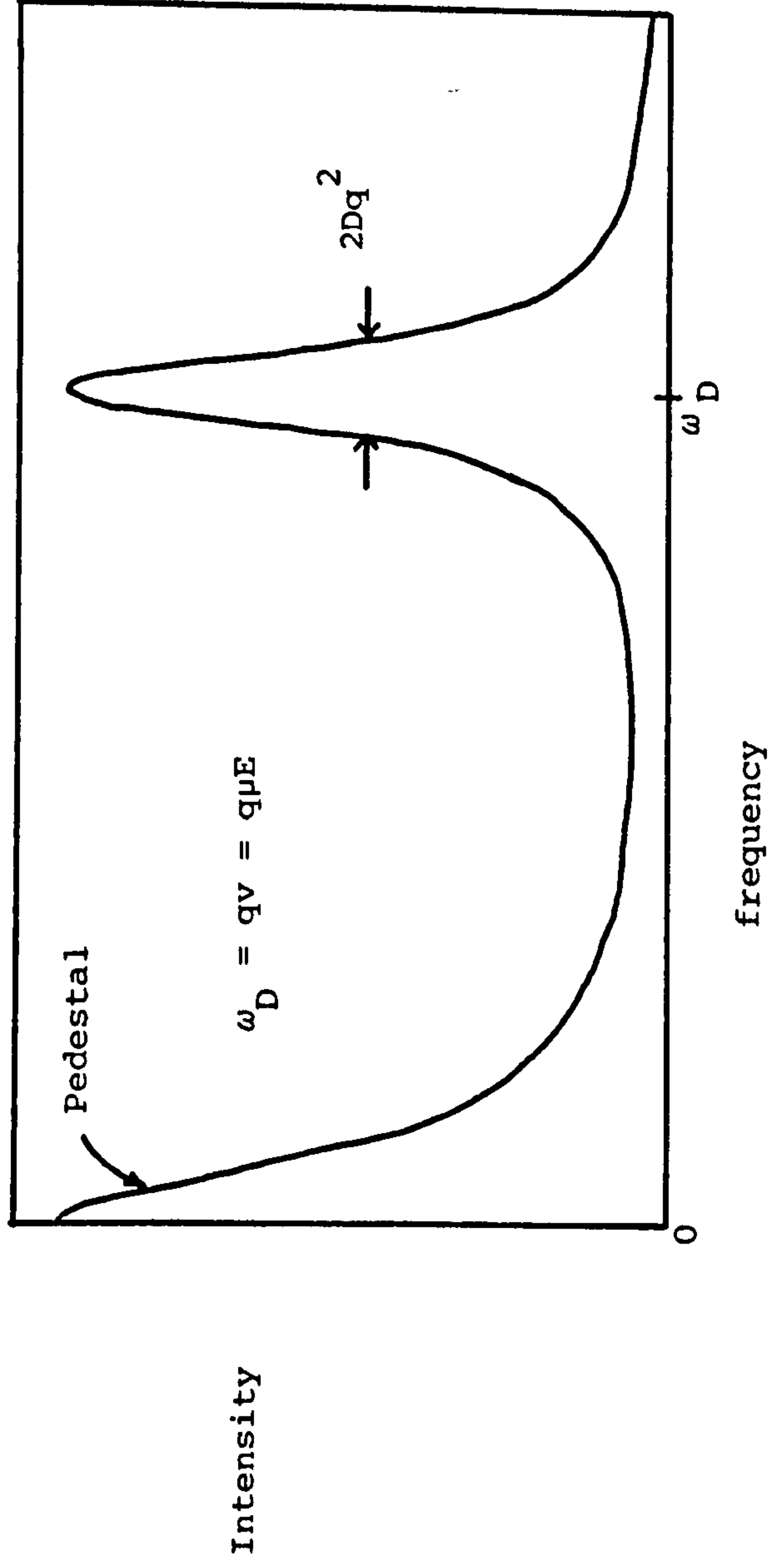


Figure 3.10: Expected frequency spectrum for a typical LDE experiment.

- (iii) relative intensities of the incident beams,
- (iv) optics alignment and light collecting equipment.

Durst and Whitelaw<sup>97</sup> provide the following equation for determining the required laser power,  $P_L$ , for use in a laser-Doppler apparatus,

$$P_L = \frac{10^4 \cdot D_I^2 \cdot h\nu \cdot \nu_D}{16 \pi^2 \eta_q \eta_c Q_{\text{scat}} r_p^2 N_{\text{ph}}} \quad (3.53)$$

where  $D_I$  is the diameter of the focussed laser beam,  $h\nu$  the photon energy,  $\nu_D$  the Doppler shift,  $\eta_c$  the light collection efficiency,  $\eta_q$  the quantum efficiency of the detector,  $Q_{\text{scat}}$  the scattering efficiency of the sample,  $r_p$  the particle radius and  $N_{\text{ph}}$  the number of fringes observed by the detector. Quantitatively, eq. (3.53) is not very useful<sup>95</sup>, but it does show that, for example, higher laser power is required for smaller, faster-moving particles.

Intuitively, it is expected that the optimum ratio of particle size to fringe spacing should be  $\sim 0.5$ , so that the motion of the particle across a fringe results in maximum change in scattered intensity. Theoretically, the optimum value has been calculated as  $0.586$ <sup>98</sup>. In practice, it is found that a wide range of particle sizes can be faithfully studied using a fixed fringe spacing.

The intensities of the incident beams should be the same in order to obtain maximum contrast between light and dark fringes. If the intensities are very different, the dark fringes will no longer be of zero intensity, but only slightly different than the light fringes. Poor "fringe visibility" leads to reduced signal to noise ratio.

Finally, incorrect alignment of the optics or badly focussed light collecting equipment will reduce the signal to noise ratio.

Typical signals obtained from LDE experiments yield pedestals with half-height half-widths of the order of 10 Hz. A typical, highly charged aqueous dispersion subjected to an electric field of  $\sim 5$  to  $10 \text{ kV m}^{-1}$  will yield Doppler frequencies of the order of 10 -100 Hz. (LDE experiments using the real fringe method usually employ scattering angles between  $10^\circ$  and  $20^\circ$ ). Thus, to avoid overlap between the two peaks, high field strengths are generally desirable, and scattering vectors should be chosen such that the diffusion broadening of the desired Doppler peak is not too great, i.e.

$$Dq^2 < q \cdot v \quad . \quad (3.54)$$

Small  $q$  values are therefore desirable, but this will require higher electric field strengths to be used in order to obtain suitable Doppler shifts.

Clearly, in order to study dispersions with small electrophoretic mobilities either high field strengths need to be used (undesirable), large scattering angles are required (reduces signal to noise ratio and increases diffusion broadening) or the pedestal needs to be removed. The latter option is generally favoured.

#### 3.5.4 PEDESTAL REMOVAL

The easiest and cheapest method available for removing the pedestal signal is to use a high-pass filter. This works well when the desired frequency component is much higher than the width of the pedestal. However, for LDE experiments, this may not always be the case, and hence simple filtering may distort the desired Doppler signal component, as well as removing the pedestal.

A more elaborate method which can remove the pedestal signal, even when the desired component lies within the pedestal band width, has been used by Bossel et al<sup>99</sup> and Dändliker and Iten<sup>100</sup>. A plane polarised laser beam is passed through a Wollaston prism, which yields two emerging beams, one at a polarisation angle of  $+45^{\circ}$  to the incident beam, and the other at  $-45^{\circ}$ . These two beams are then passed through the sample as normal and the scattered light focussed onto a second Wollaston prism. Two photodetectors are used to detect the  $+45^{\circ}$  and  $-45^{\circ}$  polarised components separately. Subtraction of the two detected signals removes the pedestal signal without distorting the required part of the signal.

A more commonly used method for removing the pedestal signal exploits the fact that the pedestal frequency occurs at an absolute frequency near to 0Hz. Thus if the Doppler signal could somehow be modulated by a high frequency, then high pass filtering and subsequent demodulation of the detected signal would remove the pedestal signal. Moreover, both positive and negative Doppler frequencies would be obtained, allowing the direction of the particle motion to be determined. The method used to modulate the Doppler signal is described below.

### 3.5.5 MODULATION OF THE DOPPLER SIGNAL BY LIGHT FREQUENCY SHIFTING

If a frequency shift  $\Delta\omega_s$  is induced in one of the incident laser beams used in a real fringe experiment, the interference fringes formed will move in the direction of  $\underline{q}$  at a rate such that  $\Delta\omega_s/2\pi$  fringes move past a fixed point in space per second. Thus, a particle moving with a velocity  $V$  in the  $\underline{q}$ -direction will yield a Doppler frequency of  $\Delta\omega_s + qv$ . A particle moving with the same speed but in the opposite direction will yield a Doppler frequency of  $\Delta\omega_s - qv$ ; i.e. the Doppler spectrum is shifted by  $\Delta\omega_s$  and the signal becomes direction-sensitive. However, the pedestal signal remains centred around 0Hz. For LDE experiments,  $\Delta\omega_s/2\pi$  is chosen to be a few kHz. For  $qv/2\pi$  typically up to 100 Hz, the detected signal can be easily high-pass filtered to remove the pedestal without any deterioration of the desired component. The filtered signal

can then be fed into a suitable spectrum analyser that can demodulate the signal at  $\sim \Delta\omega_s$ . Examples of spectra obtained in this manner are presented later.

A number of methods are available for frequency-shifting laser radiation, which are amply described by Durst et al<sup>95</sup>. The simplest and cheapest method utilises a rotating diffraction grating, or half-wave plate placed in the path of the beam. However, this does suffer from the possibility of generating unwanted mechanical vibrations in the equipment. For electrophoretic studies, this may be a problem. The method often used, and chosen for the PALS apparatus described later, is that of acousto-optic modulation. The operation of an acousto-optic modulator (AOM), or "Bragg cell", is based upon the principles of the Debye-Sears effect<sup>101</sup>. Figure 3.11 illustrates the functioning of a Bragg cell. An acoustic propagation is set up in a suitable medium, such as crystalline lead molybdate. The sound waves generate density fluctuations in the crystal, which, in turn, change the refractive index of the crystal. The incident laser beam is passed through the same crystal at such an angle that diffraction of the beam occurs due to the propagating sound wave. The output from the cell consists essentially of a zeroth order beam and two first order diffracted beams with frequencies equal to the incident laser frequency,  $\omega_o$ , plus the acoustic frequency,  $\omega_b$ , and  $\omega_o - \omega_b$ . Proper alignment results in most of the beam power being concentrated in one of the first order beams. However, the typical



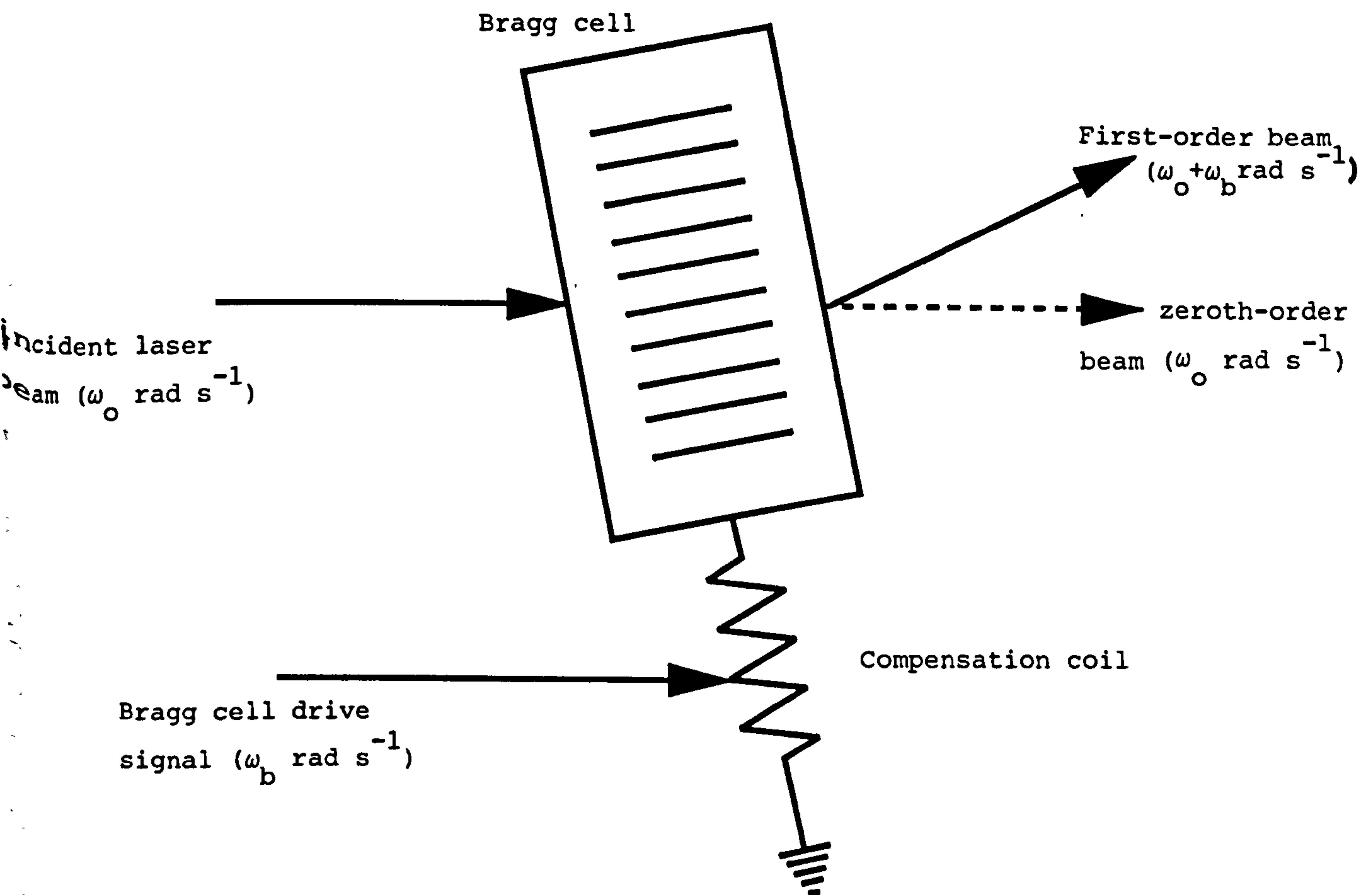


Figure 3.11: Schematic representation of the functioning of a Bragg cell (after ref. 95).



acoustic frequencies required to give good first order beams are between 40 and 100 MHz, depending on the diffracting medium. For example, the Isomet model 1205C AOM used lead molybdate as the diffracting medium and has an optimum operating frequency of 80 MHz. For electrophoresis studies, a typical "fringe" frequency between 500 Hz and 2kHz is used, hence frequency shifting one of the beams is not sufficient. It is necessary to frequency shift both of the laser beams - one by  $\omega_b$  and the other by  $\omega_b + \Delta\omega_s$ . Figure 3.12 is a schematic representation of a real fringe, direction sensitive LDE apparatus, utilising two AOMs.

Commercial Bragg cells are provided as complete units containing the suitable medium and an ultrasonic transducer to convert an external radio frequency (RF) signal into the required ultrasonic signal. Generation of the RF signals,  $\exp(i(\omega_b + \Delta\omega_s)t)$  and  $\exp(i\omega_b t)$  is described later.

### 3.5.6 PRACTICAL LIMITATIONS OF THE MOVING FRINGE LDE TECHNIQUE

This section is concerned with discussing the resolution limits for the moving fringe LDE technique employing frequency analysis of the Doppler signal, and low frequency square waves as the electrode drive signals.

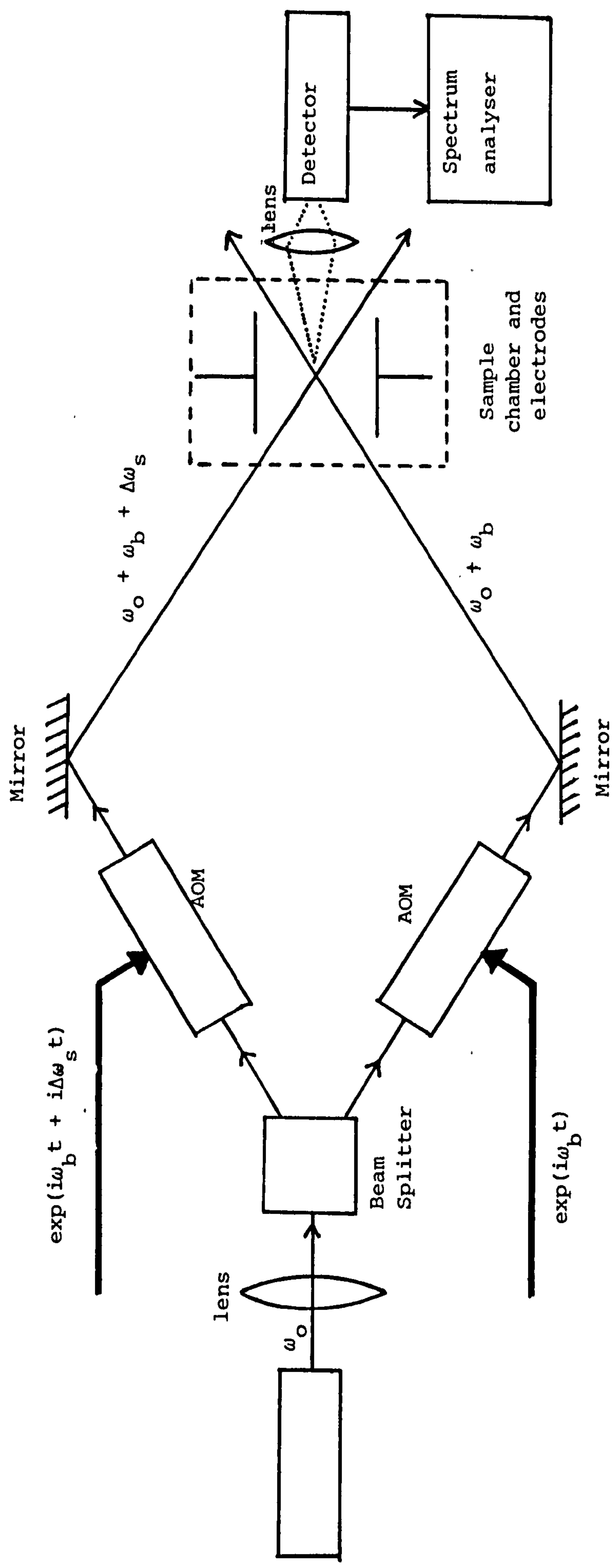
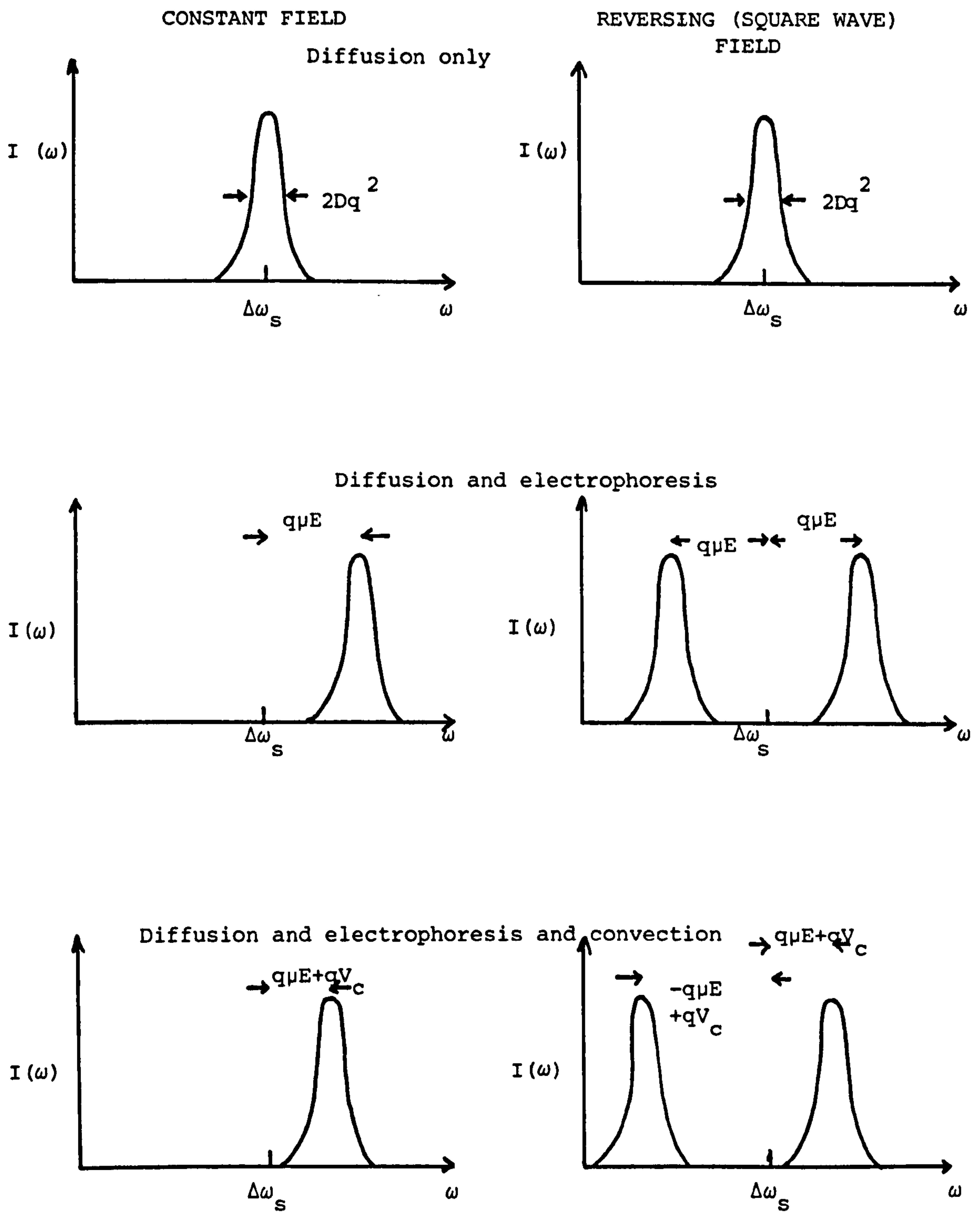


Figure 3.12: Schematic representation of a real fringe LDE apparatus incorporating moving fringes.

It has already been stated that the pedestal signal does not appear in the useful part of the Doppler spectrum when using moving fringes. The use of square wave electric fields is desirable so as to minimise the effects of polarisation and Joule heating. Although electrophoretic motion in a reversing field will yield two peaks in the Doppler spectrum for a given species (one at  $\Delta\omega_s - q\mu E$ , the other at  $\Delta\omega_s + q\mu E$ ), this is advantageous if another collective motion,  $V_c$ , due to convection for example, is present. The various schematic spectra in Figure 3.13 illustrate this. The difference between the two electrophoretic peaks arising through motion in the square wave electric field is simply  $2q\mu E$ . Neither motion in a constant field nor the static fringe LDE method can easily allow for the convective Doppler shift,  $qv_c$ . Examples of real Doppler spectra are given later.

The use of symmetric square wave fields effectively breaks the temporal averaging procedure into many short time batches. The duration of a single positive (or negative) pulse imposes an additional limit on spectral resolution. The Doppler spectrum, which was continuous for a true d.c. field, now consists of a series of harmonic peaks spaced at multiples of the applied field frequency. Thus, it is necessary to keep the frequency of the applied square wave small compared to  $q\mu E$ ; i.e. electrophoretic motion needs to occur over several fringe spacings during a single field pulse.



**Figure 3.13:** Representative spectra for an ideal dispersion , illustrating how using a square wave electric field removes ambiguities in determined  $q\mu E$  due to convection, for example.



Although this is not a critical limitation for many aqueous colloidal dispersions, it does impose severe restrictions for non-polar dispersions. High voltages need to be applied across the electrodes for systems with small mobilities. For example, an aqueous system with  $\mu = 1 \times 10^{-8} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$  can successfully be studied by LDE using an applied field of 5V across electrodes 2mm apart with a reversal rate of 2Hz. A non-polar system with  $\mu = 1 \times 10^{-10} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$  will require 500V across the same electrodes if similar Doppler shifts are to be obtained. Thus, the problem of excessive Joule heating is likely to arise (§ 2.5.1). If the restriction of multiple fringe crossing by the electrophoresing particles could be removed, then this problem would be reduced and dispersions with very small electrophoretic mobilities could be studied without the need for very strong electric fields.

### 3.6 PHASE ANALYSIS LIGHT SCATTERING<sup>82, 102, 103, 104</sup>

Phase analysis light scattering (PALS) is a modification of the moving fringe LDE technique described above which removes the restriction of multiple fringe crossing by particles undergoing electrophoresis. The PALS theory presented here is an extension of the theory for the technique described by Schätzel and Merz<sup>102</sup>.

The key difference between conventional LDE and PALS is that, in the latter case, the moving fringes are used for phase demodulation of the laser-Doppler signal. It is assumed that the frequency difference between the two laser beams is determined by an audio frequency (AF) generator yielding a "reference" signal  $\exp(i \Delta\omega_s t)$ .

### 3.6.1 RELATING PARTICLE DISPLACEMENTS TO OPTICAL PHASE CHANGES

If one considers a stationary scatterer located within the fringe volume, then the observed scattered light signal will simply be a sine wave of frequency  $\Delta\omega_s$ . There will be an arbitrary, but fixed phase shift,  $\phi_{arb}$ , between the scattered light signal and the reference signal, determined to some extent by the filter characteristics of the electronics, but, more importantly, the initial position of the scatterer,  $x_j(0)$ . If the stationary scatterer is moved by the equivalent of half a fringe spacing (i.e.  $d/2$ ) in the direction of  $\underline{q}$ , the phase shift between the reference signal and the scattered light signal will change by  $\pi$  radians. In general, for a moving scatterer,  $j$ , the phase at a time  $t$  will be given by

$$\begin{aligned}\phi_j(t) &= \phi_{arb} + q [x_j(t) - x_j(0)] \\ &= q x_j(t)\end{aligned}\tag{3.55}$$

when  $\phi_{\text{arb}}$  is taken to be  $x_j(0)$ . Figure 3.14 illustrates how the optical phase, relative to  $\exp(i \Delta\omega_s t)$  depends on the position of a single scattering particle. Thus, if a particle moves in the direction of  $\underline{q}$  with a velocity  $V$ , the phase difference will change by  $v \cdot 2\pi \cdot 2 \sin(\theta/2)/\lambda \text{ rad s}^{-1}$  (from eq. (3.51)). The phase change as a function of time will resemble that depicted in figure 3.15. Hence, the mean phase change with time should yield the collective, constant motion,  $x^{(c)}(t)$ , while the mean squared phase change should also contain a diffusion contribution,  $x_j^{(d)}(t)$ , i.e.

$$\langle \Delta \phi(t) \rangle = \frac{d x^{(c)}(t)}{dt} = v^{(c)}(t) \quad (3.56)$$

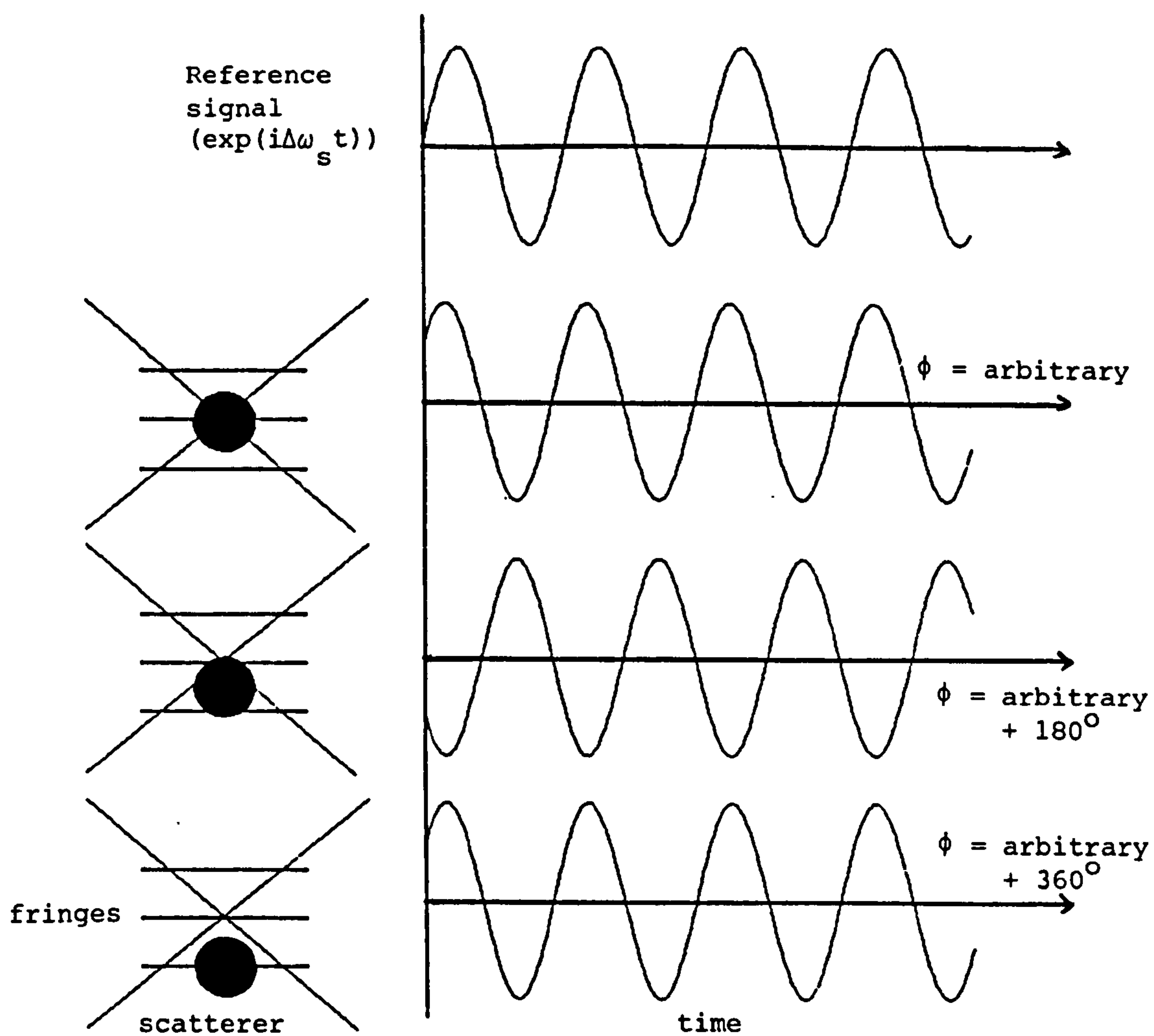
and

$$\langle \Delta \phi(t)^2 \rangle = (v^{(c)}(t))^2 + \langle [\phi_j^{(d)}(t) - \phi_j^{(d)}(0)]^2 \rangle \quad (3.57)$$

$$= (v^{(c)}(t))^2 + Dq^2 t \quad (3.58)$$

For electrophoresis in a d.c. field,  $v^{(c)}(t) = \mu E$ . There is no lower limit to the required particle displacement other than the detection noise of the apparatus. Thus, in principle, this method is ideally suited to the study of electrophoretic behaviour of non-polar colloidal dispersions. The situation is more complicated for the phase of a more realistic, many particle laser-Doppler signal. Let  $A(t)$  and  $\phi(t)$  be the amplitude and phase of the complex Doppler signal respectively (relative to  $\exp(i \Delta\omega_s t)$ ). The complex Doppler signal,  $u(t)$ , relative to  $\exp(i \Delta\omega_s t)$ , is then





**Figure 3.14:** Diagram to illustrate how the optical phase of the scattered light, relative to  $\exp(i\Delta\omega_s t)$ , depends on the position of a single scatterer.

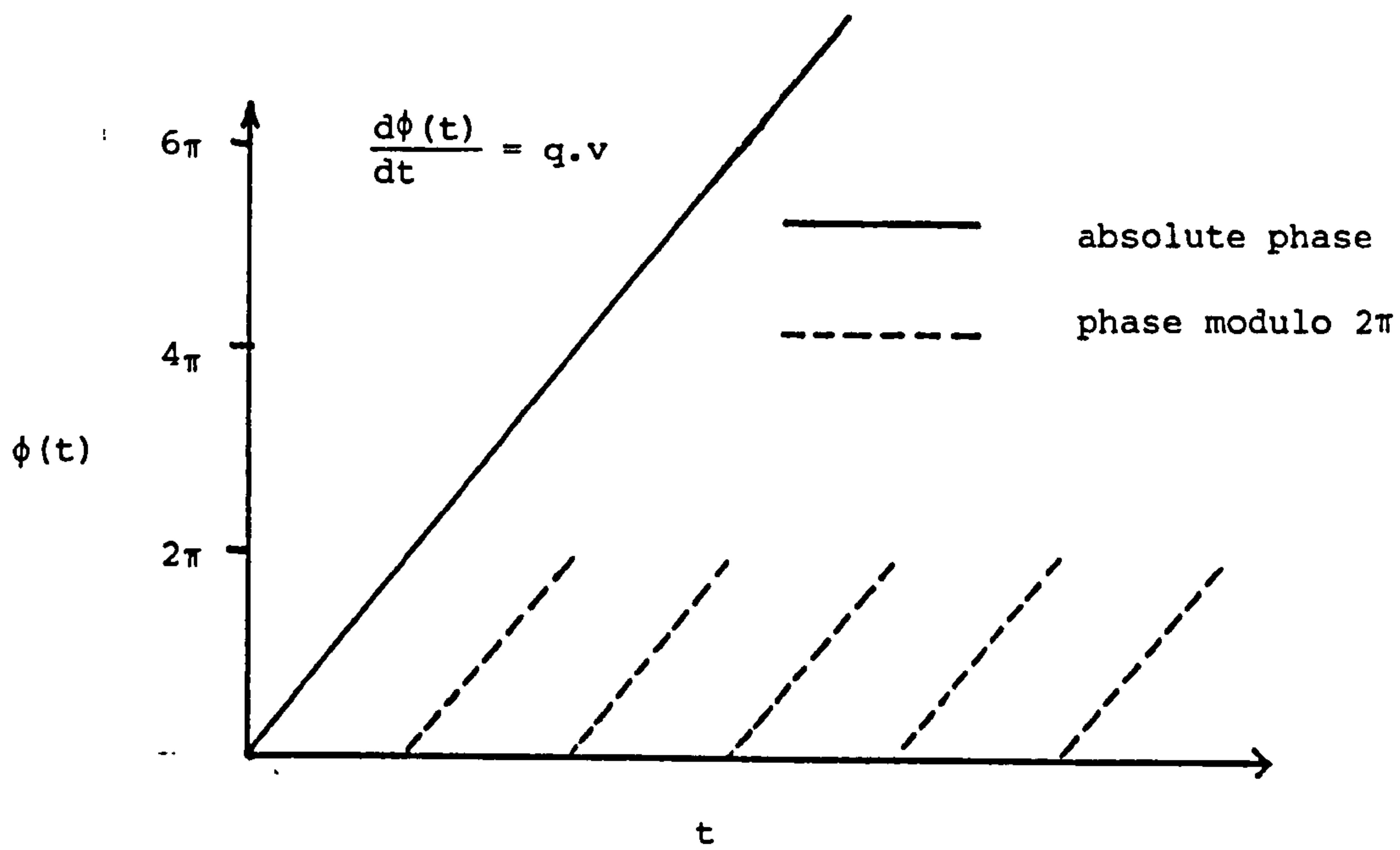


Figure 3.15: Phase change occurring for a particle moving with a uniform velocity  $V$  in the direction of  $q$ .

$$u(t) = A(t) \exp (i \phi (t)) \quad . \quad (3.59)$$

The phase contribution due to collective motion,  $\phi^{(c)}(t)$ , can be defined by

$$\dot{\phi}^{(c)}(t) = q\mu E(t) \quad (3.60)$$

and a diffusive phase contribution is represented by  $\phi^{(d)}(t)$ . For electrophoresis and diffusive contributions only,

$$\phi(t) = \phi^{(c)}(t) + \phi^{(d)}(t) \quad . \quad (3.61)$$

Electrophoretic motion is contained in  $\phi^{(c)}(t)$ , only. Diffusion does not affect  $\phi^{(c)}(t)$ , but completely determines  $A(t)$  and  $\phi^{(d)}(t)$ . The random nature of diffusion ensures that  $\langle \phi^{(d)}(t) \rangle = 0$  over some time  $\tau$ . Thus, the averaged phase change may be used to estimate the electrophoretic motion. An alternative scheme, which provides better statistical accuracy, is described below.

### 3.6.2 THE TWO-TIME DISTRIBUTION OF THE COMPLEX DOPPLER SIGNAL AND ITS IMPORTANCE

The complex Doppler signal,  $u(t)$ , relative to  $\exp (i \Delta \omega_s t)$ , represents a complex Gaussian stochastic process with independent real and imaginary parts and a temporal correlation (cf. eq. (3.20)).

$$\langle u(t) u^*(t + \tau) \rangle = \langle A^2 \rangle \exp (-q^2 D \tau) \quad . \quad (3.62)$$

The probability density distribution,  $p(\bar{x})$ , for an n-dimensional Gaussian process  $\bar{x}$  is

$$p(\bar{x}) = \frac{1}{(2\pi)^{n/2} \sqrt{(|\text{Cov}|)}} \cdot \exp \left( -\frac{1}{2} \bar{x}^T \text{Cov}^{-1} \bar{x} \right) \quad (3.63)$$

where  $\text{Cov} = (\langle x_i x_j \rangle)_{ij}$  and  $(a_{ij})^T = (a_{ji})$ . For a one dimensional Gaussian process

$$p(\bar{x}) = \frac{1}{\sqrt{(2\pi)}} \cdot \sigma \exp \left( -\bar{x}^2 / (2\sigma^2) \right) \quad (3.64)$$

The covariance of a two-dimensional process is related to  $\sigma$  by

$$\text{Cov} = \sigma^2 \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \quad (3.65)$$

The value of  $\rho$  will be seen later. The inverse covariance is

$$\text{Cov}^{-1} = \sigma^2 \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}^{-1} \quad (3.66)$$

$$= \sigma^2 \left( \begin{array}{cc|cc} 1 & \rho & 1 & 0 \\ \rho & 1 & 0 & 1 \end{array} \right) \quad (3.67)$$

$$= \sigma^2 \left( \begin{array}{cc|cc} 1 - \rho^2 & \rho - \rho^3 & 1 - \rho^2 & 0 \\ \rho - \rho^3 & 1 - \rho^2 & 0 & 1 - \rho^2 \end{array} \right) \quad (3.68)$$

$$= \sigma^2 \left( \begin{array}{cc|cc} 1 - \rho^2 & \rho - \rho^3 & 1 - \rho^2 & 0 \\ 0 & 1 - 2\rho^2 + \rho^4 & -\rho + \rho^3 & 1 - \rho^2 \end{array} \right) \quad (3.69)$$

$$= \sigma^2 \left( \begin{array}{cc|cc} 1 & -\rho^2 & \rho(1-\rho^2) & 1-\rho^2 & 0 \\ 0 & (1-\rho^2)^2 & -\rho(1-\rho^2) & -\rho(1-\rho^2) & (1-\rho^2) \end{array} \right) \quad (3.70)$$

$$= \sigma^2 \left( \begin{array}{cc|cc} 1 & -\rho^2 & 0 & 1 & -\rho \\ 0 & 1-\rho^2 & -\rho & -\rho & 1 \end{array} \right) \quad (3.71)$$

For the Doppler signal, let  $x$  and  $y$  be the real and imaginary components; i.e.

$$x + iy = A \exp(i\phi) \quad (3.72)$$

and, at a time  $\tau$  later,

$$x' + iy' = A' \exp(i\phi') \quad (3.73)$$

The 2-time probability distribution function of  $u(t)$  at  $t=t$  and  $t=t+\tau$  is

$$p(x, y, x', y') = p(x, y) \cdot p(x', y') \quad (3.74)$$

Now,

$$p(x, y) = \frac{1}{2\pi\sigma^2\sqrt{(1-\rho^2)}} \exp\left\{-\frac{1}{2}(x, y) \begin{pmatrix} 1 & -\rho \\ \rho & 1 \end{pmatrix} \frac{1}{\sigma^2(1-\rho^2)} \begin{pmatrix} x \\ y \end{pmatrix}\right\} \quad (3.75)$$

$$= \frac{1}{2\pi\sigma^2\sqrt{(1-\rho^2)}} \exp\left\{-\frac{1}{2\sigma^2(1-\rho^2)} \left[ (x-\rho y, -\rho x+y) \begin{pmatrix} x \\ y \end{pmatrix} \right]\right\} \quad (3.76)$$

$$= \frac{1}{2\pi\sigma^2\sqrt{(1-\rho^2)}} \exp\left\{-\frac{1}{2\sigma^2(1-\rho^2)} [x^2 - \rho xy - \rho xy + y^2]\right\} \quad (3.77)$$

Hence,

$$p(x, y, x', y') = \frac{1}{4 \pi^2 \sigma^4 (1-\rho^2)} \exp\left\{-\frac{1}{2 \sigma^2 (1-\rho^2)} \left[ x^2 - 2\rho xy + y^2 + x'^2 - 2\rho x' y' + y'^2 \right]\right\} \quad (3.78)$$

Now,

$$\langle x^2 \rangle = \sigma^2 \quad (3.79)$$

and

$$\langle A^2 \rangle = \langle x^2 \rangle + \langle y^2 \rangle = 2\sigma^2 \quad (3.80)$$

From eq. (3.61)

$$\sigma^2 \rho = \langle xy \rangle = \sigma^2 \exp(-Dq^2 \tau) \quad (3.81)$$

thus,

$$\rho = \exp(-Dq^2 \tau) . \quad (3.82)$$

Hence, the two-time distribution of  $u(t)$  reads

$$p(x, y, x', y') = \frac{1}{(2\pi)^2 \sigma^4 (1-\rho^2)} \exp\left\{-\frac{1}{2\sigma^2 (1-\rho^2)} \left[ A^2 + A'^2 - 2\rho A A' \cos(\phi - \phi') \right]\right\} \quad (3.83)$$

$$= A A' \exp \{ -[A^2 + A'^2 - 2 A A' \exp (-q^2 D\tau) \cos (\phi' - \phi)] \} / [\langle A^2 \rangle (1 - \exp (-2q^2 D\tau))] \} / [\pi^2 \langle A^2 \rangle^2 (1 - \exp (-2q^2 D\tau))] \quad (3.84)$$

= p (A, δA, φ, δφ), where A = A(t), A' = A + δA = A (t+τ) and φ = φ<sup>(d)</sup> (t), φ' = φ + δφ = φ<sup>(d)</sup> (t+τ).

For small τ, exp (-q<sup>2</sup> Dτ) ≈ 1 - q<sup>2</sup> Dτ, hence the 2-time distribution simplifies to

$$p (A, \delta A, \phi, d\phi) \approx A A' \exp \{ -[A^2 + A'^2 - 2AA' (1 - q^2 D\tau) \cos (\delta\phi)] \} / [\langle A^2 \rangle 2q^2 D\tau] / [\pi^2 \langle A^2 \rangle^2 (2q^2 D\tau)] \quad (3.85)$$

$$\approx A^2 \exp \{ -[2A^2 + 2A\delta A + \delta A^2 - 2A (A + \delta A) (1 - q^2 D\tau) (1 - \delta\phi^2/2)] \} / [\langle A^2 \rangle 2q^2 D\tau] \} / [\pi^2 \langle A^2 \rangle^2 (2q^2 D\tau)] \quad (3.86)$$

$$\approx A^2 \exp \{ -[\delta A^2 + 2A^2 q^2 D\tau + A^2 \delta\phi^2] \} / [\langle A^2 \rangle 2q^2 D\tau] \} / [\pi^2 \langle A^2 \rangle^2 (2q^2 D\tau)] \quad (3.87)$$

$$\approx (2A / \langle A^2 \rangle) \exp \{ -A^2 / \langle A^2 \rangle \} \cdot (1/2\pi) \cdot [2\pi \langle A^2 \rangle (q^2 D\tau)]^{-\frac{1}{2}} \exp \{ -\delta A^2 / [2 \langle A^2 \rangle q^2 D\tau] \} \cdot A [2\pi \langle A^2 \rangle (q^2 D\tau)]^{-\frac{1}{2}} \exp \{ -A^2 \delta\phi^2 / [2 \langle A^2 \rangle q^2 D\tau] \} \quad (3.88)$$

Eq. (3.88) is a product of 4 distributions: a Rayleigh distribution for the amplitude, A, a uniform distribution for the initial phase, φ ,



and Gaussian distributions for the amplitude change,  $\delta A$ , and for the product of amplitude and phase change,  $A\delta\phi$ . The individual random variables for each of the probability density distribution functions in eq. (3.88) are statistically independent.  $A\delta\phi$  is statistically independent of  $A$ ,  $\phi$  and  $\delta A$ , but not  $\delta\phi$  itself. The distribution of phase changes,  $\delta\phi$ , has a width which is proportional to the inverse of  $A$ . Hence, as  $A$  tends to zero, the variance term, and all higher moments, of  $\delta\phi$  diverges. This is as one would expect, since the phase of a signal with zero amplitude is ill-defined. Scattering from many particles will occasionally result in destructive interference, leading to very small amplitudes. Hence, the use of mean square phase changes for determining velocity information from many-particle signals is undesirable. The diverging variance of  $\delta\phi$  indicates that the distribution of  $\delta\phi$  decays slowly, resulting in poor statistical accuracy for mean phase changes, too.

From eq. (3.88) the distribution of  $\delta\phi$  may be computed as a Student's distribution with two degrees of freedom,

$$p(\delta\phi) = q^2 D\tau (2q^2 D\tau + \delta\phi^2)^{-3/2}, \quad (3.89)$$

whereas the distribution for the random variable  $A\delta\phi$  is a Gaussian with variance  $\langle A^2 \rangle q^2 D\tau$ .  $A\delta\phi$  is defined as a new variable,  $\delta Q$ ,

$$\delta Q = A \cdot \delta\phi \quad (3.90)$$

Hence,

$$p(\delta Q) = \left[ 2\pi \langle A^2 \rangle (q^2 D \tau) \right]^{-1/2} \exp \left\{ -\delta Q^2 / [2\langle A^2 \rangle q^2 D \tau] \right\} \quad (3.91)$$

$\delta Q$  is termed the "amplitude weighted phase change". Figure 3.16 shows the probability densities for the diffusive part of unweighted and amplitude weighted phase changes for  $2q^2 D \tau = 1$ . The sharper decay of  $p(\delta Q)$  indicates that the use of amplitude weighted phase changes for the determination of velocity information will be statistically superior than the use of unweighted phase changes.

### 3.6.3 THE AMPLITUDE-WEIGHTED PHASE DIFFERENCE (AWPD) FUNCTION

A finite time, amplitude-weighted phase difference (AWPD) can be defined by temporal integration over  $\delta Q$ ,

$$\begin{aligned} Q(t) - Q(0) &= \int_0^t A(t') \dot{\phi}(t') dt' \\ &\approx \sum_{j=0}^{t/\tau} A(j\tau) \delta\phi(j\tau) . \end{aligned} \quad (3.92)$$

For diffusive motion only, the AWPD averages to zero. With incorporation of a collective motion due to electrophoresis, the AWPD reads

$$Q(t) - Q(0) = Q^{(d)}(t) - Q^{(d)}(0) + Q^{(c)}(t) - Q^{(c)}(0) , \quad (3.93)$$

$$Q^{(c)}(t) - Q^{(c)}(0) = \int_0^t A(t') \dot{\phi}^{(c)}(t') dt'$$

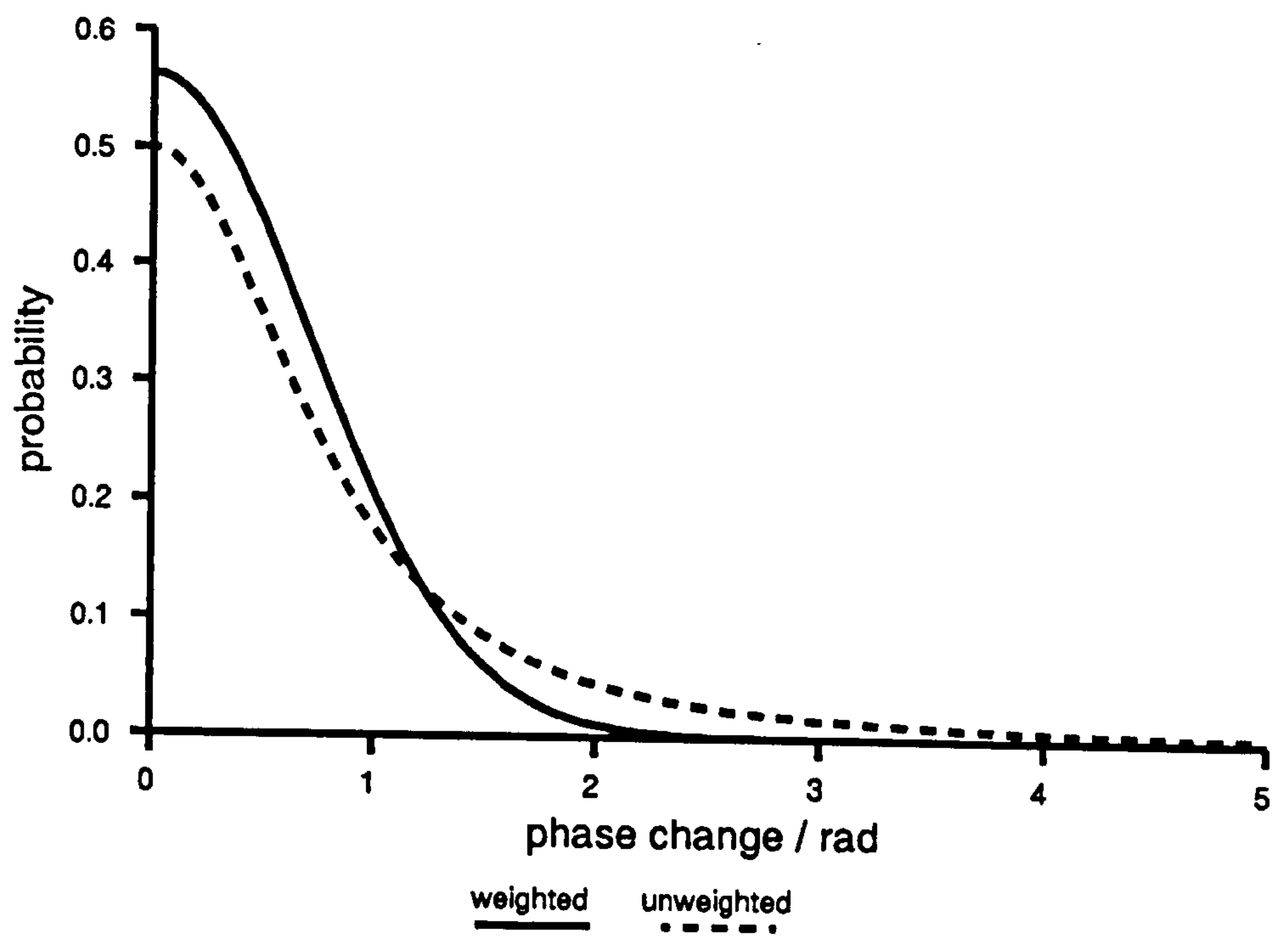


Figure 3.16: Probability densities for diffusive part of unweighted and weighted phase changes for  $2\eta^2 Dt = 1$ .

$$= \int_0^t A(t') q \mu E(t') dt'. \quad (3.94)$$

The mean value of the AWPD may be obtained by ensemble averaging, or, more conveniently, by temporal averaging over many identical cycles of the applied field,  $E(t)$ , yielding

$$\langle Q(t) - Q(0) \rangle = \langle A \rangle q \mu \int_0^t E(t') dt'. \quad (3.95)$$

For a sinusoidal field,

$$E(t) = E_0 \sin(\omega_e t + \phi), \quad (3.96)$$

eq. (3.95) becomes

$$\begin{aligned} \langle Q(t) - Q(0) \rangle &= \langle A \rangle q \mu E_0 \int_0^t \sin(\omega_e t' + \phi) dt' \\ &= \langle A \rangle q \mu E_0 \{ \cos \phi - \cos(\omega_e t + \phi) \} / \omega_e. \end{aligned} \quad (3.97)$$

Since  $\phi$ ,  $\omega_e$  and  $E_0$  are known by design of the experiment, and  $\langle A \rangle$  may be determined easily from the measured Doppler signal, a measurement of the mean AWPD yields the wanted electrophoretic mobility.

The above account of the AWPD function has assumed that only diffusive motion and electrophoretic motion under the influence of a sinusoidal electric field are present in the sample. In many real cases, another collective motion may be present due to, for example,

thermal convection or sedimentation/creaming. If this collective motion is described by a constant velocity,  $V_c$ , then the AWPB becomes

$$\langle Q(t) - Q(0) \rangle = \langle A \rangle q \left[ v_c t + \mu E_0 \{ \cos \phi \cos (\omega_e t + \phi) \} / \omega_e \right] \quad (3.98)$$

Hence, a PALS experiment generating AWPB functions can readily determine both the electrophoretic mobility of a sample and a constant velocity term,  $v_c$ . The signs of both velocity components are obtained.

#### 3.6.4 THE AMPLITUDE-WEIGHTED PHASE STRUCTURE (AWPS) FUNCTION

It has already been stated that a random process  $x(t)$  can be described by its temporal correlation function (§ 3.3). It is also possible to describe it by the mean square change over some time  $\tau$ ,

$$S(\tau) = \langle |x(t) - x(t - \tau)|^2 \rangle \quad (3.99)$$

$S(\tau)$  is known as the structure function of  $x(t)$ <sup>103</sup>. Thus, the amplitude-weighted phase structure function (AWPS), first described by Schätzel and Merz<sup>102</sup>, is

$$\begin{aligned} \langle [Q(t' + t) - Q(t')]^2 \rangle &= \langle [Q^{(d)}(t+t') - Q^{(d)}(t')]^2 \rangle \\ &+ \langle [Q^{(c)}(t+t') - Q^{(c)}(t')]^2 \rangle \end{aligned} \quad (3.100)$$



The first term is simply  $\langle A^2 \rangle q^2 Dt$ , since  $\delta Q(t)$  is a Gaussian random variable. The collective (electrophoretic) term is

$$\begin{aligned} \langle [Q^{(c)}(t+t') - Q^{(c)}(t')]^2 \rangle &= q^2 \mu^2 \int_0^t \int_0^t \langle E(t') E(t'') \rangle dt' dt'' \\ &= \langle A(t') A(t'') \rangle dt' dt'' . \end{aligned} \quad (3.101)$$

For  $t \gg q/Dq^2$ ,  $\langle A(t') A(t'') \rangle$  may be replaced by its long time limit,  $\langle A \rangle^2$ . Hence, for electrophoretic motion in a sinusoidal field, the collective AWPS term is

$$\begin{aligned} \langle [Q^{(c)}(t+t') - Q^{(c)}(t')]^2 \rangle &= q^2 \mu^2 E_0^2 \langle A \rangle^2 \cdot \\ &\quad \int_0^t \int_0^t \cos(\omega_e t' - \omega_e t'') dt' dt'' / 2 \\ &= q^2 \mu^2 E_0^2 \langle A \rangle^2 [1 - \cos \omega_e t] / \omega_e^2 . \end{aligned} \quad (3.102)$$

Including a constant velocity,  $v_c$ , the total AWPS reads

$$\begin{aligned} \langle [Q(t'+t) - Q(t')]^2 \rangle &= \langle A^2 \rangle q^2 Dt + q^2 \langle A \rangle^2 \{ v_c^2 t^2 + \\ &\quad \mu^2 E_0^2 [1 - \cos \omega_e t] / \omega_e^2 \} . \end{aligned} \quad (3.103)$$

Hence, the AWPS function can be used to obtain the diffusion coefficient, electrophoretic mobility and the constant velocity of the dispersion being studied. Note, however, that in contrast to the AWPD function, the AWPS function does not yield the sign of either  $\mu$  or  $v_c$ .

The necessary relation between  $\langle A \rangle$  and  $\langle A^2 \rangle$  is easily computed from the Rayleigh distribution of  $A$  in eq. (3.88) as

$$\langle A^2 \rangle = 4 / \pi \cdot \langle A \rangle^2 . \quad (3.104)$$

It is important to note that the AWPS function is obtained by continuous averaging over all times,  $t'$ , whereas the AWPD function is obtained by synchronous averaging with cycles of the applied field. If, however, the mean square amplitude-weighted phase change is also obtained by synchronous averaging with the applied field, then eq. (3.103) becomes more complicated. The continuously-averaged AWPS function may be expressed as

$$\langle [Q(t+\tau) - Q(t)]^2 \rangle = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T [Q(t+\tau) - Q(t)]^2 dt , \quad (3.105)$$

whereas, for the synchronously-averaged AWPS function,

$$\langle [Q(t+\tau) - Q(t)]^2 \rangle = \lim_{T \rightarrow \infty} \frac{1}{M} [Q(t+\tau) - Q(t)]^2 \Big|_{t=TM+t_0} \quad (3.106)$$

where  $M$  is the number of electrode cycles of period  $T$ , and  $t_0$  is the time on the electrode cycle at which averaging starts. Thus, for a sinusoidal motion,  $\hat{X} \sin(\omega_e t)$  ( $\hat{X} = \mu E_0 / \omega_e$ ), and a constant velocity motion,  $v_c t$ ,



$$\langle [Q^{(c)}(t+\tau) - Q^{(c)}(t)]^2 \rangle = \langle A \rangle^2 q^2 [V_c(t_0 + \tau) - V_c t_0 + \hat{X} \sin(\omega_e t_0 + \omega_e \tau) - \hat{X} \sin(\omega_e t_0)]^2 \quad (3.107)$$

$$= \langle A \rangle^2 q^2 [V_c^2 \tau^2 + 2V_c \tau \hat{X} (\sin(\omega_e t_0 + \omega_e \tau) - \sin(\omega_e t_0))] + \hat{X}^2 [(\sin(\omega_e t_0 + \omega_e \tau) - \sin(\omega_e t_0))^2] \quad (3.108)$$

$$= \langle A \rangle^2 q^2 [V_c^2 \tau^2 + 2V_c \tau \hat{X} (2 \cos((2\omega_e t_0 + \omega_e \tau)/2)) \cdot \sin(\omega_e \tau/2) + \hat{X}^2 (2 \cos((2\omega_e t_0 + \omega_e \tau)/2) \sin(\omega_e \tau/2))^2] \quad (3.109)$$

$$= \langle A \rangle^2 q^2 [V_c^2 \tau^2 + 4V_c \tau (\mu E_0 / \omega_e) \cos((2\omega_e t_0 + \omega_e \tau)/2) \cdot \sin(\omega_e \tau/2) + (\mu E_0 / \omega_e)^2 \cdot (1 + \cos(2\omega_e t_0 + \omega_e \tau)) \cdot (1 - \cos(\omega_e \tau))] \quad (3.110)$$

Eq. (3.110) contains a cross term with the constant and sinusoidal motions, as well as the independent  $V_c^2 \tau^2$  and  $(\mu E_0 / \omega_e)^2$  terms.

Averaging over all  $t_0$  reduces eq. (3.110) to

$$\left\langle \left| Q^{(c)}(t+\tau) - Q^{(c)}(t) \right|^2 \right\rangle_{t_0} = \langle A \rangle^2 q^2 \left[ v_c^2 \tau^2 + (\mu E_0 / \omega_e)^2 (1 - \cos(\omega_e \tau)) \right] \quad (3.111)$$

which is the same as for the continuously averaged AWPS function.

Figure 3.17 illustrates some typical APWD and AWPS functions for both synchronous and continuous averaging.

#### 3.6.5 AWPD vs. AWPS FUNCTIONS FOR ELECTROPHORETIC MOBILITY DETERMINATION

With two different methods available for determining the electrophoretic mobility by PALS, either via the AWPS or AWPD functions, it is necessary to assess the relative merits of each. A comparison of the statistical accuracy of both methods is given in Appendix 1.

In wishing to interpret the electrophoretic mobilities of non-polar dispersions, it is clearly very important to obtain both the magnitude and the sign of the mobility. Hence, the AWPD function is preferred. However, the AWPS function can provide diffusive information about the sample, which may also be of importance. The diffusion coefficient of the particles may also be obtained from the correlation functions described below (§ 3.6.7 and 3.6.8). As will be seen from the AWPD and AWPS functions for various samples studied for this work, excessive diffusion of the sample can dominate the AWPS function. For

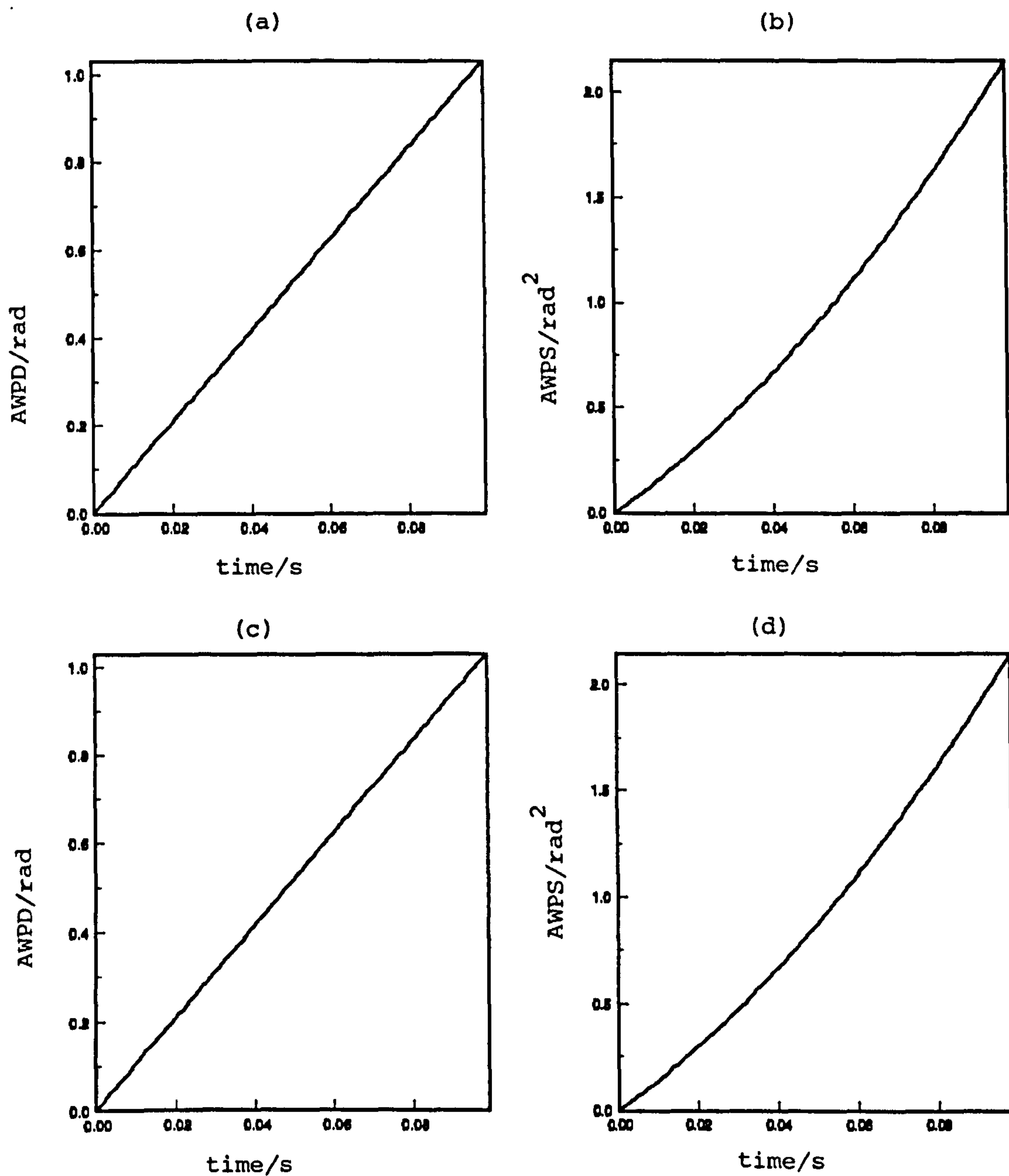


Figure 3.17: Theoretical AWP and AWPS functions for a 500nm radius spherical particle dispersed in water at 298K. Scattering angle =  $15.2^\circ$  in air. Laser wavelength = 632.8 nm.  $V_c = 2 \mu\text{m s}^{-1}$ .

- |     |             |      |
|-----|-------------|------|
| (a) | Synchronous | AWPD |
| (b) | Synchronous | AWPS |
| (c) | Continuous  | AWPD |
| (d) | Continuous  | AWPS |

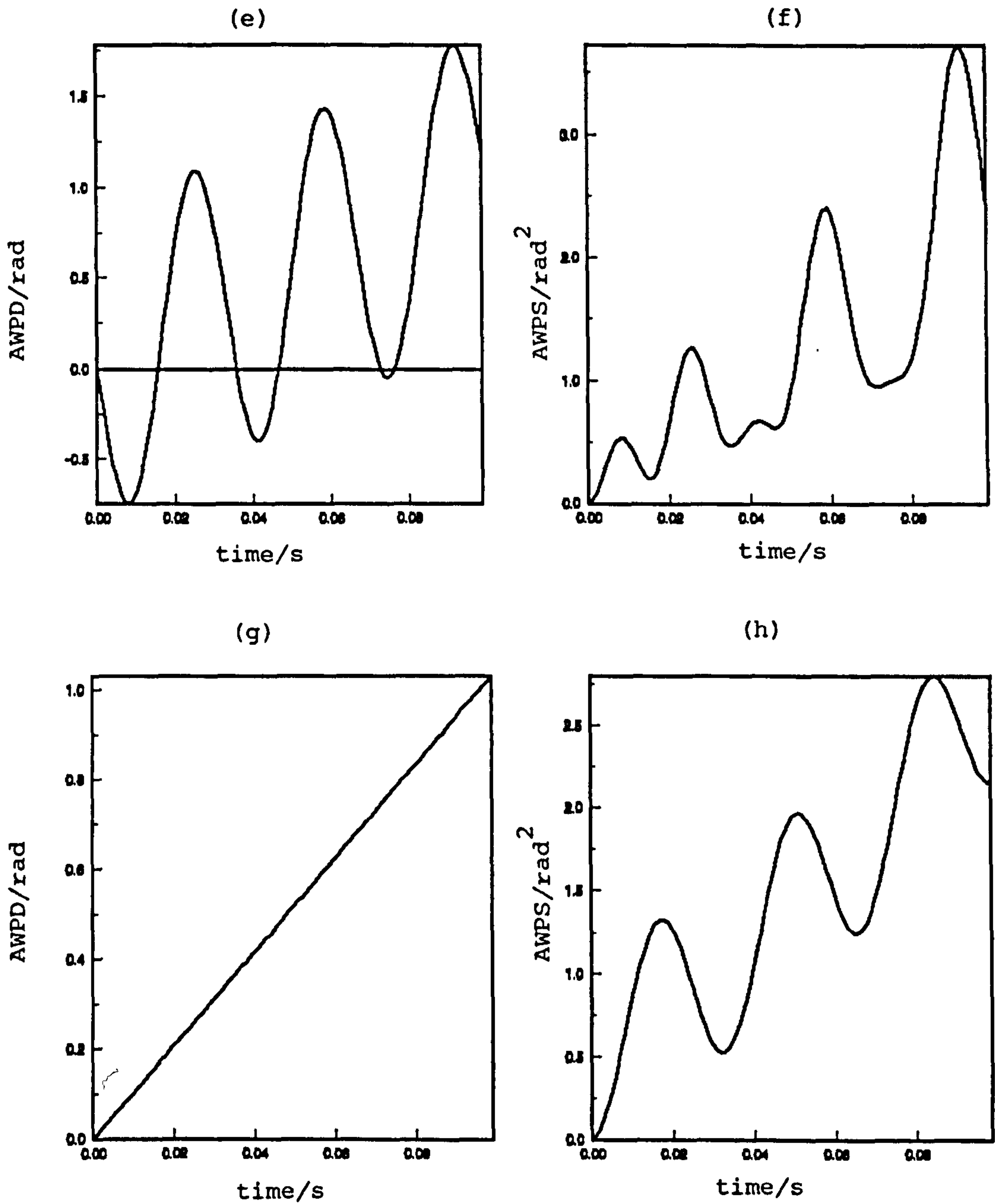


Figure 3.17: Theoretical AWP and AWPS functions for a 500 nm radius spherical particle dispersed in water at 298 K. Scattering angle =  $15.2^\circ$  in air. Laser wavelength = 632.8 nm.  $V = 2 \mu\text{m s}^{-1}$ .  $\mu = 3 \times 10^{-8} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$ ,  $E = 100 \text{ V mm}^{-1}$  at 30.0 Hz.

- (e) Synchronous AWP
- (f) Synchronous AWPS
- (g) Continuous AWP
- (h) Continuous AWPS

systems with very small  $\mu$ , the electrophoretic component of the AWPS function is often undetectable. However, by using the AWPD function, the electrophoretic component is not dominated by diffusion since the diffusion term averages to zero.

Although high resolution determination of the electrophoretic term in the AWPD function is not diffusion limited, it may be limited by an excessive constant velocity term,  $v_c$ . Ideally, the constant velocity term should be as small as possible. However, sedimentation or thermal convection of the sample can lead to constant velocities higher than  $20 \mu\text{m s}^{-1}$ . For a sample with  $\mu = 5 \times 10^{-11} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$ , subjected to a sinusoidal electric field of 100 V across 2mm at a frequency of 30 Hz, the maximum displacement of the particles due to the field will be of the order of 100 nm. Thus, to determine  $\mu$ , the linear component of the AWPD function needs to be minimised. A method for achieving this is described below. Examples illustrating the importance of this method are presented later.

### 3.6.6 MINIMISATION OF THE CONSTANT VELOCITY TERM

The scattered light signal obtained from the detector,  $U_o(t)$ , during a PALS experiment is



$$U_o(t) = \sum_{j=1}^N a_j(t) \exp[ i \phi_j(t) + i \Delta \omega_s t ] . \quad (3.112)$$

Expanding  $\phi_j(t)$  into the relevant diffusive, electrophoretic and constant velocity terms,  $\phi_j^{(d)}(t)$ ,  $\phi^{(e)}(t)$  and  $\phi^{(v)}(t)$ , respectively, yields

$$U_o(t) = \sum_{j=1}^N a_j(t) \exp [ i \phi_j^{(d)}(t) + i \phi^{(e)}(t) + i \phi^{(v)}(t) + i \Delta \omega_s t ] . \quad (3.113)$$

Hence, if the amplitude and phase information required for the AWPD (or AWPS) function are obtained by demodulation of  $U_o(t)$  relative to a frequency  $\phi^{(v)}(t) + \Delta \omega_s$ , instead of  $\Delta \omega_s$ , then the complex signal processed to obtain the AWPD (or AWPS) function will be

$$U(t) = \sum_{j=1}^N a_j(t) \exp [ i \phi_j^{(d)}(t) + i \phi^{(e)}(t) ] . \quad (3.114)$$

Therefore, determination of the AWPD function will yield electrophoretic information only, and the corresponding AWPS function will yield electrophoretic and diffusive information only. The value of the appropriate demodulation frequency may be obtained by using a spectrum analyser, since the constant velocity term simply results in a linear displacement of the Doppler signal in the frequency domain (eq. (3.40)). An alternative method is described later, and, as will be seen, proves to be very useful for studying dispersions with extremely low electrophoretic mobilities.

### 3.6.7 CORRELATION FUNCTIONS FOR THE REAL/IMAGINARY PART OF THE COMPLEX DOPPLER SIGNAL

From the previous discussions of autocorrelation functions used in PCS, it is clear that autocorrelation of the real or imaginary component of the PALS complex Doppler signal, relative to  $\exp(i \Delta\omega_s t)$ , will yield the first-order autocorrelation function. For diffusive contributions only, this will be the same as eq. (3.20). A relationship exists between the first-order correlation function and the structure function for a stationary, random process,  $x(t)$ ,<sup>103</sup>

$$G(\tau) = \langle x(t) x(t-\tau) \rangle \quad (3.115)$$

$$S(\tau) = \langle [x(t) - x(t-\tau)]^2 \rangle \quad (3.116)$$

$$\text{and} \quad S(\tau) = 2G(0) - 2G(\tau) \quad (3.117)$$

Thus, in the presence of electrophoresis in a sinusoidal field, the general nature of  $g^{(1)}(\tau)$  will depend upon whether or not synchronous averaging is employed. With continuous averaging, a sinusoidal component with frequency  $\omega_e$  is expected in the correlation function. With synchronous averaging, sinusoidal components at  $\omega_e$  and  $2\omega_e$  are expected, just as for the AWPS functions.



The recent study of Gaigalas et al<sup>75</sup> into the electrophoretic response of submicron particles to alternating fields has already been mentioned (§ 2.5.2). They used PCS to obtain homodyne autocorrelation functions for particles moving under the influence of an applied sinusoidal electric field. They found, for  $\omega_e / 2 \pi > 100$  Hz, a single sinusoidal component in the autocorrelation function. At lower frequencies, a second harmonic component is included in the correlation function. Their conclusion is that a frequency-dependent, apparent complex electrophoretic mobility is required to explain their observations. It is further suggested that the non-linear response of the particle velocity to field frequency may be caused by the inherently non-linear hydrodynamic interactions between the particles and the surrounding liquid. As will be seen from AWPD functions presented later, the electrophoretic mobility in a sinusoidal field is not frequency dependent. Gaigalas et al mistakenly assume that the homodyne experiment can yield absolute phase information for the scattered light, whereas the phase information is lost at the detector. What they observe with their experiment is a relative displacement of sinusoidal particle motion, i.e. electrokinetic polydispersity. Schätzel<sup>105</sup> has considered this problem, and obtains the following expression for the normalised second-order autocorrelation function for the experiment performed by Gaigalas et al,

$$g^{(2)}(\tau) = 1 + \exp(-2Dq^2\tau) \cdot \exp(-Z) \cdot I_0(Z) \quad (3.117)$$

$$\text{where } Z = q^2 \sigma^2 (1 - \cos(\omega_e \tau))/2 \quad (3.118)$$

$$\sigma = \Delta \hat{x} = \Delta \mu E_0 / \omega_e \quad (3.119)$$

and  $I_0(Z)$  is the Bessel function

$$I_0(Z) = \sum_{k=0}^{\infty} (Z/2)^{2k} / (k!)^2 \quad (3.120)$$

$\Delta \mu$  is the width of the distribution of electrophoretic mobilities, which gives an indication of the electrokinetic polydispersity.

### 3.6.8 CORRELATION FUNCTIONS FOR THE AMPLITUDE OF THE COMPLEX DOPPLER SIGNAL

The correlation of the amplitude of  $u(t)$  is<sup>102</sup>

$$\langle A(t) A(t') \rangle = 2\pi \int_0^{2\pi} d\phi \int_0^{\infty} A dA \int_0^{\infty} A' dA' p(A, \delta A, \phi, \delta \phi). \quad (3.121)$$

The probability distribution  $p(A, \delta A, \phi, \delta \phi)$  is given in eq. (3.84).

Thus, using  $2\sigma^2 = \langle A^2 \rangle$  and  $\rho = \exp(-Dq^2\tau)$ ,

$$\langle A(t) A(t') \rangle = \frac{1}{\sigma^2(1-\rho^2)} \int_0^\infty \int_0^\infty A^2 A'^2 \exp[ -(A^2 + A'^2)/2\sigma^2\sqrt{(1-\rho^2)}] \cdot$$

$$I_0(\rho A A' / (\sigma^2 \sqrt{(1-\rho^2)})) dA dA'$$

$$= \sqrt{(\pi/(2\sigma^2\sqrt{(1-\rho^2)}))} \int_0^\infty A'^2 \exp(-A'^2/(2\sigma^2\sqrt{(1-\rho^2)})) \cdot$$

$${}_1F_1(3/2; 1; \rho^2 A'^2 / (2\sigma^2\sqrt{(1-\rho^2)})) dA'$$

$$= (\pi/4) \cdot 2\sigma^2\sqrt{(1-\rho^2)} \cdot F(3/2, 3/2; 1, \rho^2) \quad (3.122)$$

where  ${}_1F_1$  and  $F$  denote the confluent and ordinary geometric functions, respectively. According to Schätzel<sup>105</sup>, eq. (3.122) can be approximated by

$$\langle A(t) A(t') \rangle \cong (\pi/4) \langle A^2 \rangle [1 - ((4/\pi) - 1) (0.92 \exp(-2Dq^2\tau) + 0.077 \exp(-4Dq^2\tau))] \cdot \quad (3.123)$$

Thus, the normalised amplitude correlation function is

$$\frac{\langle A(t) A(t') \rangle}{\langle A^2 \rangle} \cong (\pi/4) + (1 - (\pi/4)) \cdot 0.92 \exp(-2Dq^2\tau) + 0.077 \exp(-4Dq^2\tau) \cdot \quad (3.124)$$

Determination of the amplitude correlation function provides a further means for obtaining diffusive information about the sample under investigation.

### 3.6.9 PROBABILITY DENSITY FUNCTIONS FOR THE AMPLITUDE OF THE COMPLEX DOPPLER SIGNAL

From eq. (3.88) it is clear that a probability density function for the amplitude of the PALS signal exists,

$$p(A) = 2A/\langle A^2 \rangle \cdot \exp \{ -A^2/\langle A^2 \rangle \} \quad (3.125)$$

for a sample of identical particles.

However, the presence of large particles, such as dust, may produce spikes in the histogram at higher amplitudes. Construction of the amplitude histogram during a PALS experiment is useful for assessing the general quality of the sample. It will be seen later, though, that a "dust rejection" facility can be implemented in the apparatus.

### 3.7 CHAPTER SUMMARY

This chapter has described how laser light scattering can be used to study the dynamic behaviour of a colloidal dispersion. The limitations of the classical LDE technique have been discussed and an alternative signal processing scheme, based upon that of Schätzel and Merz<sup>102</sup>, has been described that enables very small electrophoretic mobilities to be measured without recourse to high electric fields. The advantages of the PALS technique over Schätzel and Merz's original technique are discussed later.



## CHAPTER FOUR

### INSTRUMENT DEVELOPMENT AND VALIDATION

#### 4.1 BASIC STATIC FRINGE VELOCIMETER

The PALS apparatus described herein is an extension of a static fringe differential heterodyne velocimeter originally developed by Parkins<sup>106</sup>, who based his optical arrangement upon that of Preece and Luckman<sup>107</sup>. Figure 4.1 illustrates the optical arrangement used in the original apparatus.

##### 4.1.1 OPTICAL COMPONENTS

The reasoning behind the choice of components by Parkins has been extensively described by him<sup>106</sup>, hence it is only necessary to list the components used in the basic apparatus.

All of the optical components are pin mounted on a 2 metre aluminium optical rail (Ealing Beck 22-6928) using sliding carriers. The rail is attached to a polished slate slab ( $193 \times 30.5 \times 2.5 \text{ cm}^3$ ) using levelling cross feet (Ealing Beck 22-7108). The slate bed, in turn, rests upon two partially inflated rubber innertubes. The slate bed and rubber tubes isolate the velocimeter from unwanted environmental vibrations.

The following list summarises

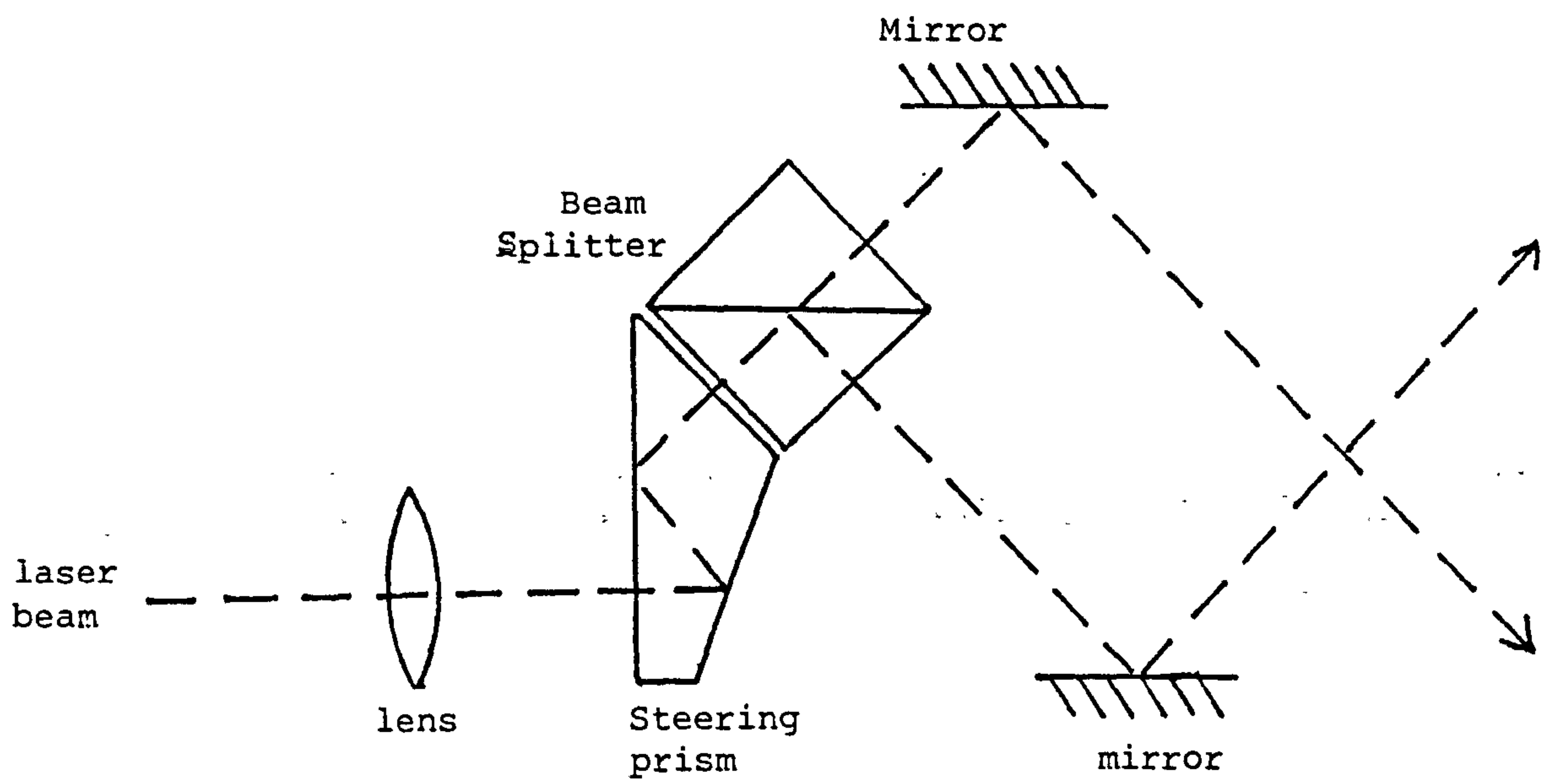


Figure 4.1: Schematic of optical arrangement used for the basic static fringe velocimeter.



the optical components comprising the original apparatus.

3225-PC He-Ne laser, plane polarised, 5.4 mW,  
 $\lambda_o = 632.8 \text{ nm}$  (Hughes Corporation), powered by Barr and  
Stroud E1103 supply;

Neutral density filter	Ealing Beck	35-3701
Laser alignment mount	"	22-1739
Transverse slide carriers	"	22-4873
500mm biconvex research quality lens	"	23-9095
Sliding grip lens holder	"	22-8106
Transverse and vertical slide carrier	"	22-4899
Steering mirrors	"	23-5853
45° prism	"	24-3824
20mm beamsplitter cube	"	24-3931
Filter holder	"	34-8847
Slide carriers	"	22-4634

The laser produces a beam of diameter 0.83mm at its front mirror and is operated in the TEM<sub>00</sub> mode (i.e. single beam, Gaussian intensity cross-section). The 500mm focal length lens serves to focus the beam at the intersection point with a diameter of 0.42mm, thereby ensuring that the resultant fringes are parallel, and non-divergent.

#### 4.1.2 PHOTODETECTOR

Parkins used a Malvern RF 313 photomultiplier (PM) assembly, fitted with a reflex viewer, in conjunction with a Hoya 80-205mm zoom lens and 25-85 mm extension tubes. This allows the image of the crossed beams to be enlarged and focussed onto a 200  $\mu$ m diameter pinhole mounted inside the PM assembly. The photomultiplier tube (PM tube) itself is a Thorn-EMI 9863 KB/100 with an S20 photocathode. Figure 4.2 represents the basic construction of a photomultiplier tube. The principle of operation of a PM tube is as follows : a high voltage ( $\sim$  - 1.7 kV) is applied across the photocathode and the anode. A photon hitting the photocathode results in the emission of an electron from the cathode. This electron is focussed onto a series of electrodes (the dynode chain). As the electron accelerates towards the anode, amplification of the current occurs. The current detected at the anode is linearly proportional to the intensity of the light falling on the cathode. The efficiency of the S20 photocathode at 632.8nm is approx. 5%. Such PM tubes fulfill the requirements mentioned in § 3.3 for application in dynamic light scattering techniques. The Malvern RF313 assembly contains electronics for conditioning the anode signal for use by digital correlators. The raw, unamplified anode signal is available at a "conhex" socket. The amplification and processing of the anode signal are described later.

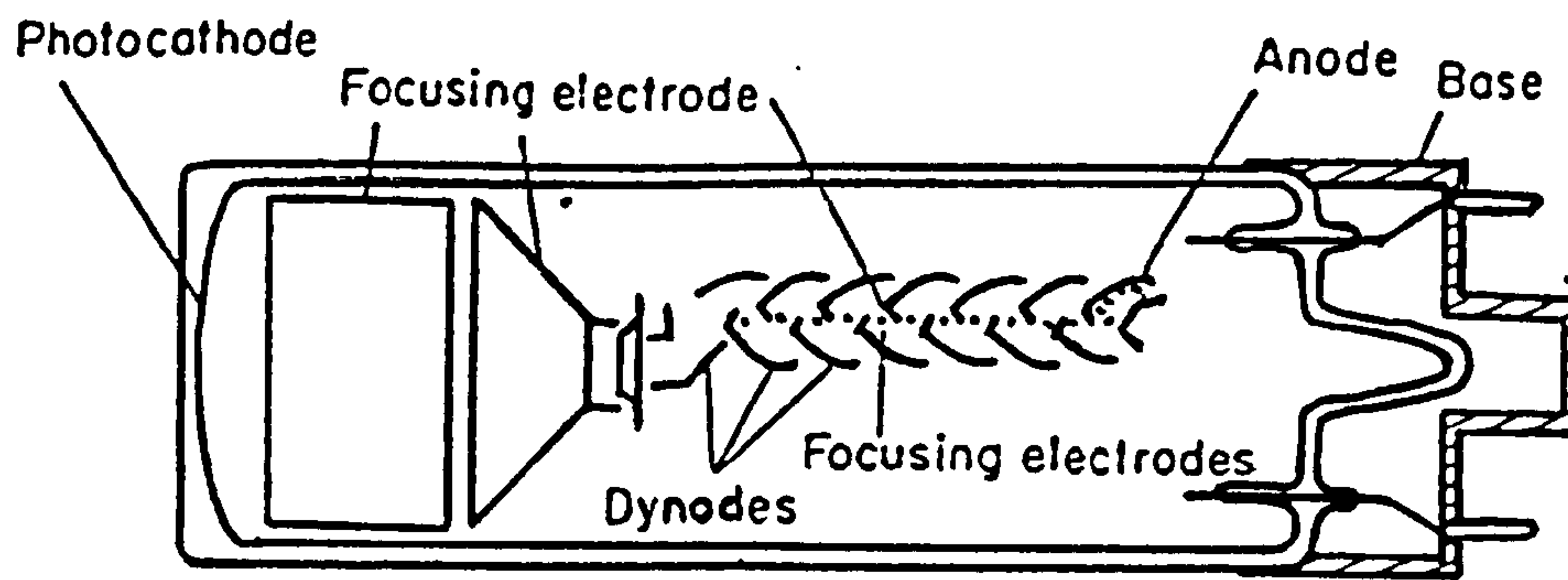


Figure 4.2: Schematic representation of a photomultiplier tube.

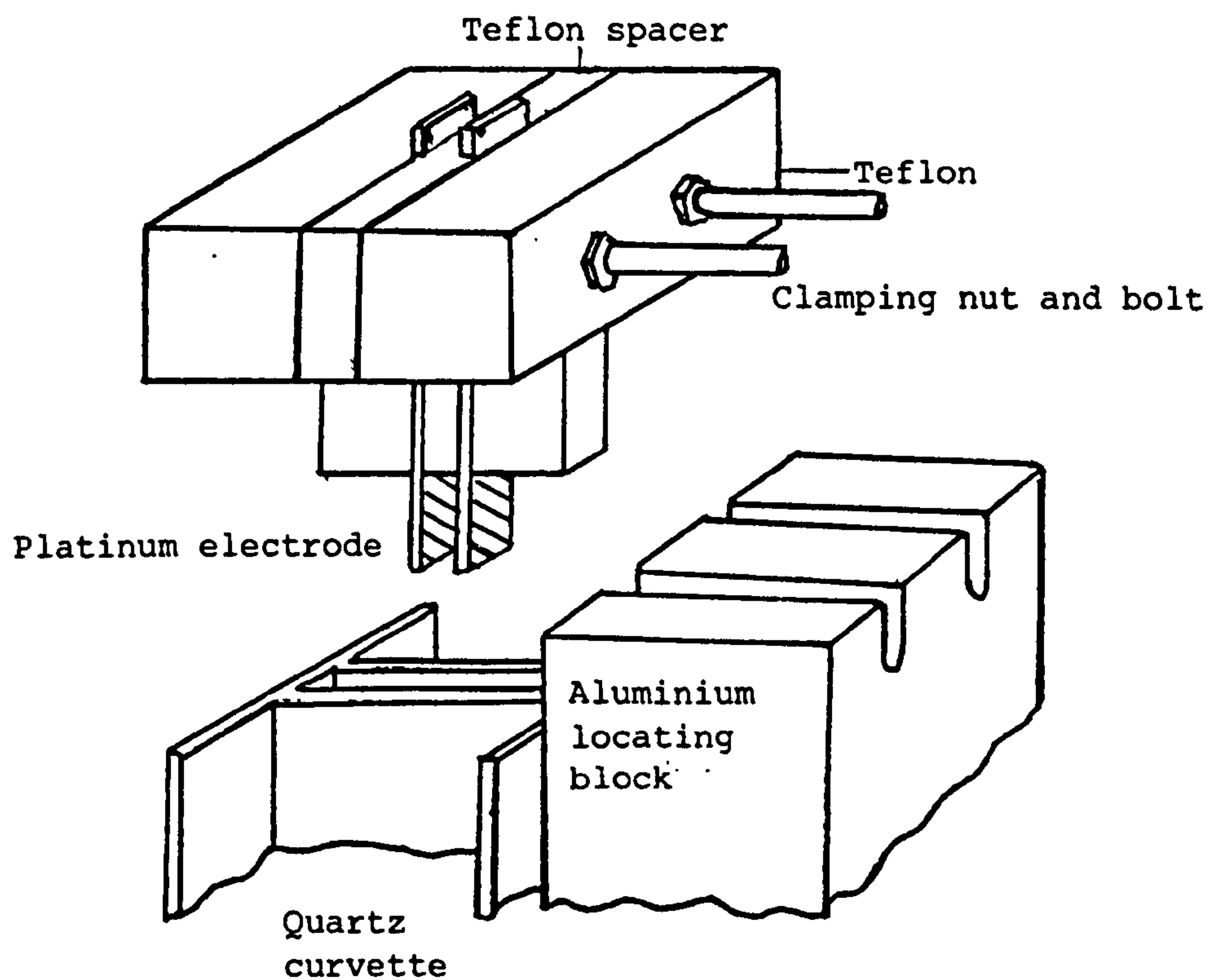


Figure 4.3: Narrow gap electrode arrangement (after Uzgis (108)).

#### 4.1.3 ELECTRODE ASSEMBLY

The electrode assembly design used throughout this work is that of Uzgiris<sup>108</sup>. Figure 4.3 illustrates the narrow gap electrode arrangement. The cuvettes used in this work were 4mm path length standard spectrophotometric cuvettes. The electrodes were made of platinum strips (ex. Johnson-Matthey), typically  $0.25 \times 2 \times 50 \text{ mm}^3$ . For aqueous electrolyte systems, where electrode reactions may result in gas formation, the platinum was "blackened" using 1% w/w chloroplatinic acid solution in conjunction with a suitable current supply<sup>106</sup>. The deposition of platinum black onto the electrode surface increases the capacitance of the electrodes thereby reducing the risk of electrolysis. For non-polar dispersions, with small conductivities, non-blackened platinum electrodes were employed. The electrode spacing (typically 1 - 4mm) can be adjusted by changing the thickness of the PTFE blocks used to construct the assembly. The electrode separation can be determined using a clock gauge (0.01mm resolution) mounted on a stationary block near to the electrode housing, in conjunction with a cross hair eyepiece mounted on the photomultiplier assembly. The generation of the electrode drive signals is described in §4.2.3. According to Uzgiris, this type of electrode assembly offers a number of useful features :

- i) it is easily dismantled, allowing quick sample changes and easy cleaning,
- ii) it allows high field strengths to be generated using small voltages,



and iii) electroosmotic flow inside the gap is eliminated.

The reduction in electroosmotic velocity by the use of a.c. fields and deep sample cells has been discussed already (§ 2.5.2). Parkins has confirmed that Uzgiris's design produces constant uniform electrophoretic velocity profiles between the electrodes.

#### 4.1.4 SAMPLE CELL TEMPERATURE CONTROL

The original velocimeter design employed a simple water jacket to provide thermostating of the sample cell. This is illustrated in figure 4.4. However, it was found that this design produced undesirable vibrations, caused by the water flowing into the jacket and striking the inner wall (point A in figure 4.4). For the PALS processing, this was found to cause excessive interference. Thus, the water jacket design was replaced by an aluminium block, with a copper coil running through to provide good heat exchange. Small pieces of card, wrapped in aluminium foil, were used to prevent heat radiation transfer through the front and rear cuvette windows, so as to minimise convective turbulence. The circulation equipment consisted of a Grant Instruments W6 water bath, KA/TA heater/circulator and CZ1 dip cooler to provide thermostatted water, and a Charles Austen Pumps Limited CP16-C pump to circulate the water around the thermostating block, illustrated in figure 4.5. Vibrations induced by the water flow through this design were found to be negligible.

Cuvette and electrode  
assembly

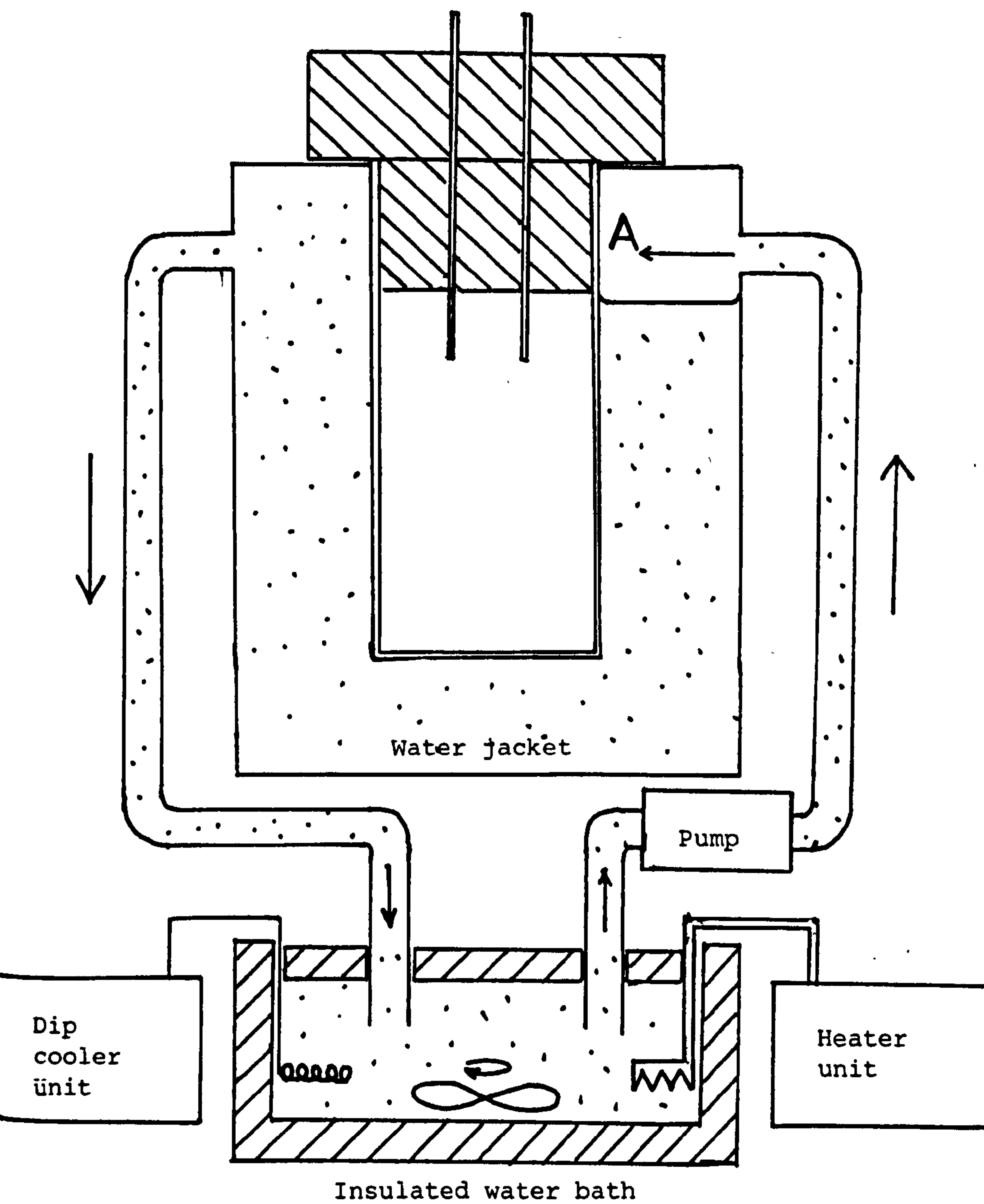


Figure 4.4: Water thermostating arrangement as used by Parkins<sup>106</sup>.



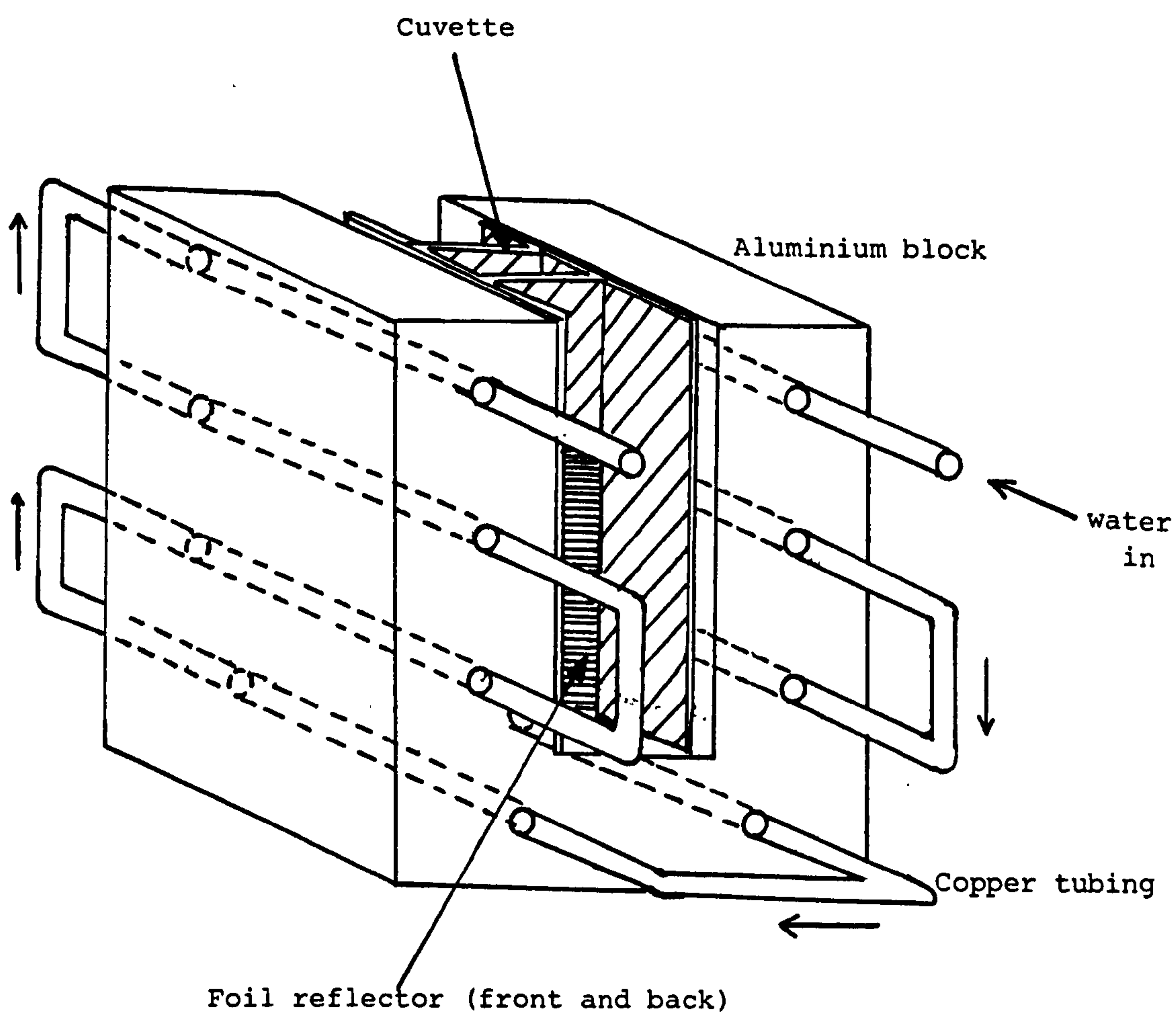


Figure 4.5: Aluminium thermostating block and cuvette (electrodes not shown for clarity).

#### 4.1.5 SIGNAL PROCESSING

The first stage of development of the basic velocimeter was the inclusion of a real-time spectrum analyser and the interfacing of the analyser to a microcomputer. The spectrum analyser used with the velocimeter is a Hewlett Packard 3582A real-time fast Fourier Transform (FFT) instrument. This provides an operating bandwidth of 0 to 25 kHz with a possible resolution of 1/256 Hz. It also provides input sensitivity ranging from 3 mV to 30V. An IEEE interface is provided, allowing data (time domain or frequency domain) to be transferred to a microcomputer. In this case, the microcomputer is a Hewlett Packard Vectra PC (an IBM PC/AT compatible). A simple program, written in "GW-BASIC", has been developed to allow frequency spectra, or time domain records, to be read from the analyser and stored on floppy disc for later analysis or printing.

As mentioned above, the lowest input sensitivity for the 3582A spectrum analyser is 3mV. Typical PM tube anode signals range in intensity from 300 nV to 300  $\mu$ V, depending on the scattering properties of the sample under investigation. Hence, amplification of the anode signal is required before the signal is processed by the spectrum analyser. Filtering of the signal is also required in order to help minimise unwanted high frequency noise and the pedestal signal, where possible. Thus, a high quality, programmable band pass filter and

amplifier unit should be employed. A Barr and Stroud EF5 Electronic Filter System was chosen. This consists of two EF5-01 programmable high pass/low pass filter units operated in series. Each unit provides an 8-pole Butterworth filter, programmable over a range of 0.01 Hz to 9.99 kHz, and an amplification of 0dB or +20 dB (x10). Thus amplification by a factor of 100 and flexible filtering of the anode signal is readily achieved. Figure 4.6 illustrates the salient features of the processing hardware for the basic velocimeter, employing frequency analysis of the scattered light signal. Verification of the accuracy of the velocity information obtained from the basic instrument has already been performed by Parkins. Further verification is given later in the section concerned with the improved apparatus, both for frequency domain and phase domain processing techniques.

#### 4.1.6 GENERAL PERFORMANCE OF THE STATIC FRINGE VELOCIMETER

Figure 4.7 shows time domain signals for ~ 500 nm diameter silica particles dispersed in aqueous  $10^{-3}$  mol dm $^{-3}$  Na Cl solution. Each trace represents a 1 second period of amplified and low-pass filtered PM tube anode signal. The upper trace (4.7(a)) represents the scattered light signal arising from the particles undergoing random diffusion only. Some high frequency noise is also present. The lower trace (4.7(b)) exemplifies how the scattered light intensity varies sinusoidally as particles traverse the fringe region due to the

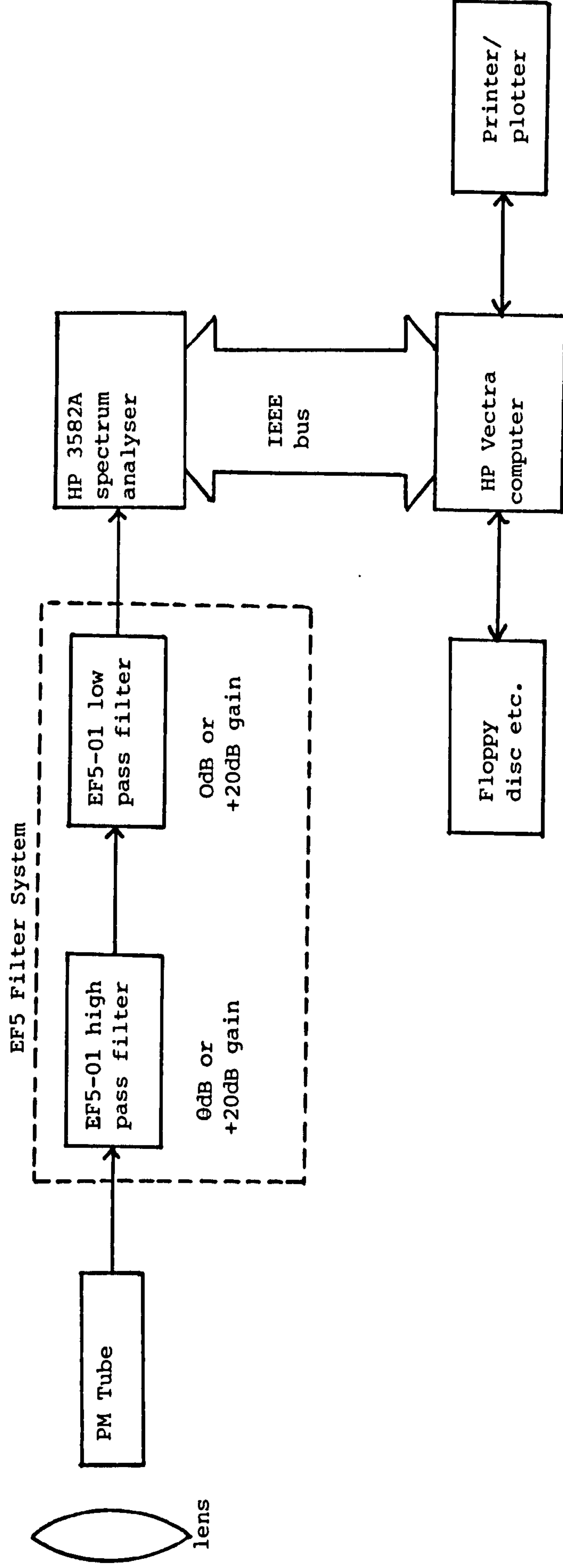


Figure 4.6: Basic electronic processing/storage equipment for static fringe velocimeter operating in frequency domain.

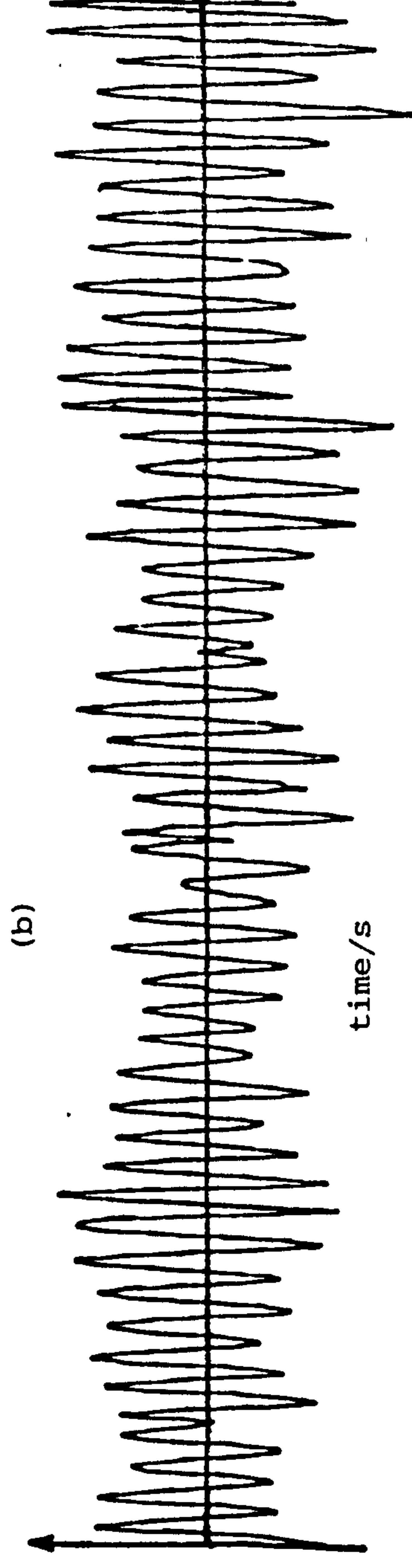
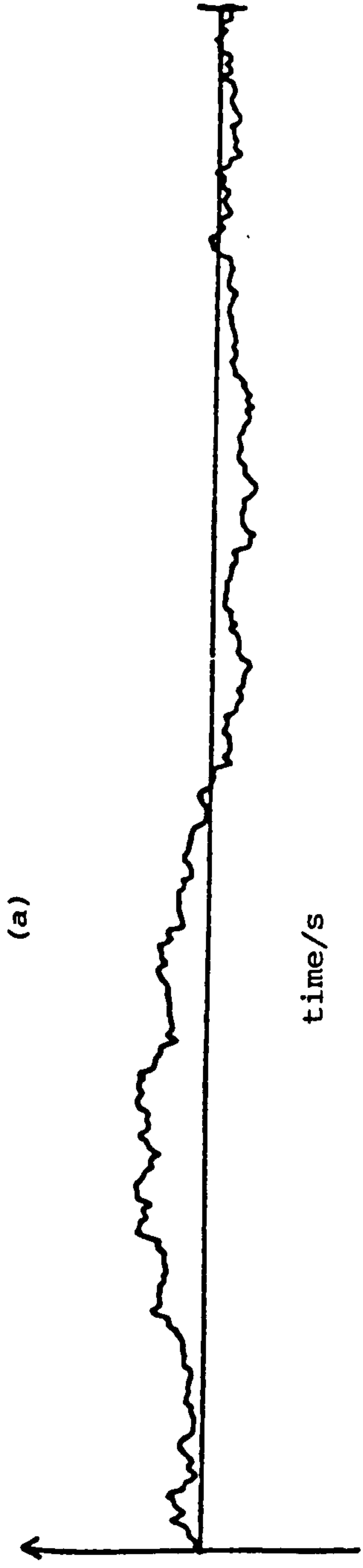


Figure 4.7: Time domain signals from basic static fringe velocimeter for ~ 500 nm diameter silica particles dispersed in aqueous  $10^{-3}$  mol  $\text{dm}^{-3}$  NaCl solution.

(a) no field applied

(b) applied field = 3.5 V square wave across 1.6 mm at 2 Hz.



application of an external field. In this case the electric field was a 3.5 V square wave, of frequency 2 Hz, applied across an electrode spacing of 1.6 mm. The corresponding frequency spectrum obtained from the experiment generating the signal typified in figure 4.7 (b) is shown in figure 4.8. In this case, the electrophoretic velocity is easily determined. Diffusion broadening is not too severe and the pedestal signal has been easily removed. However, from experience, it has been found that obtaining Doppler spectra of such quality requires obtaining the correct dilution factor (typically 1 to 10 particles in the fringe volume) and an optimum field strength. Too low a field strength makes pedestal removal difficult, whereas high field strengths often result in bubble formation in aqueous electrolytes and convective turbulence. Figure 4.9 represents more typical frequency spectra obtained for aqueous systems. The two traces show the spectra obtained with and without the electric field applied. In this case, the pedestal signal is very dominant. High field strengths are therefore needed in order to resolve the electrophoretic information.

In general, the basic static fringe velocimeter was found to work adequately for a wide range of aqueous dispersions yielding mobilities (of undetermined sign) in agreement with values obtained from commercial instruments. (A comparison between the apparatus developed for this work and a commercial instrument is given later).



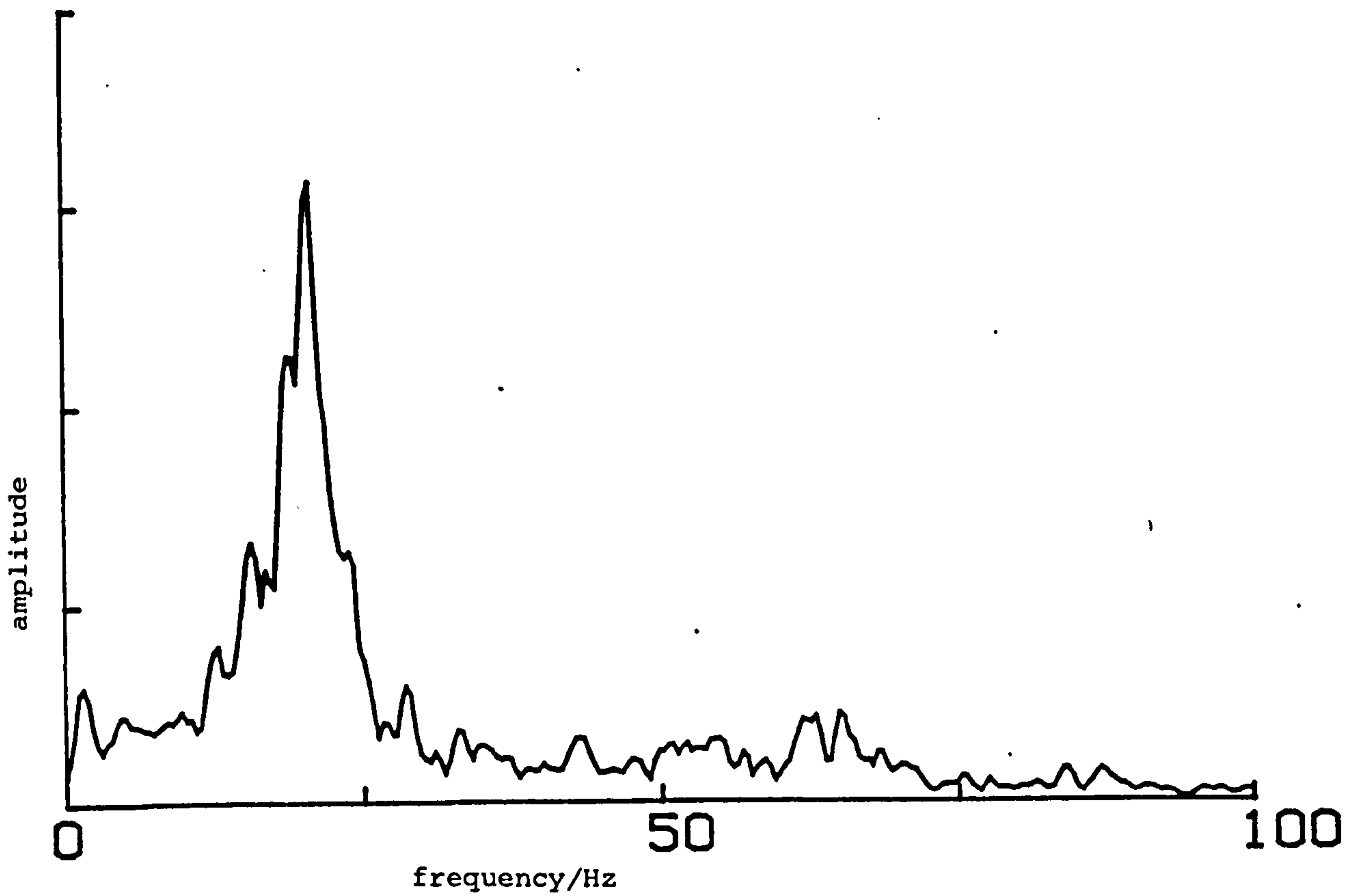


Figure 4.8: Frequency spectrum obtained for the experiment generating time time domain signal typified in figure 4.7 (b).

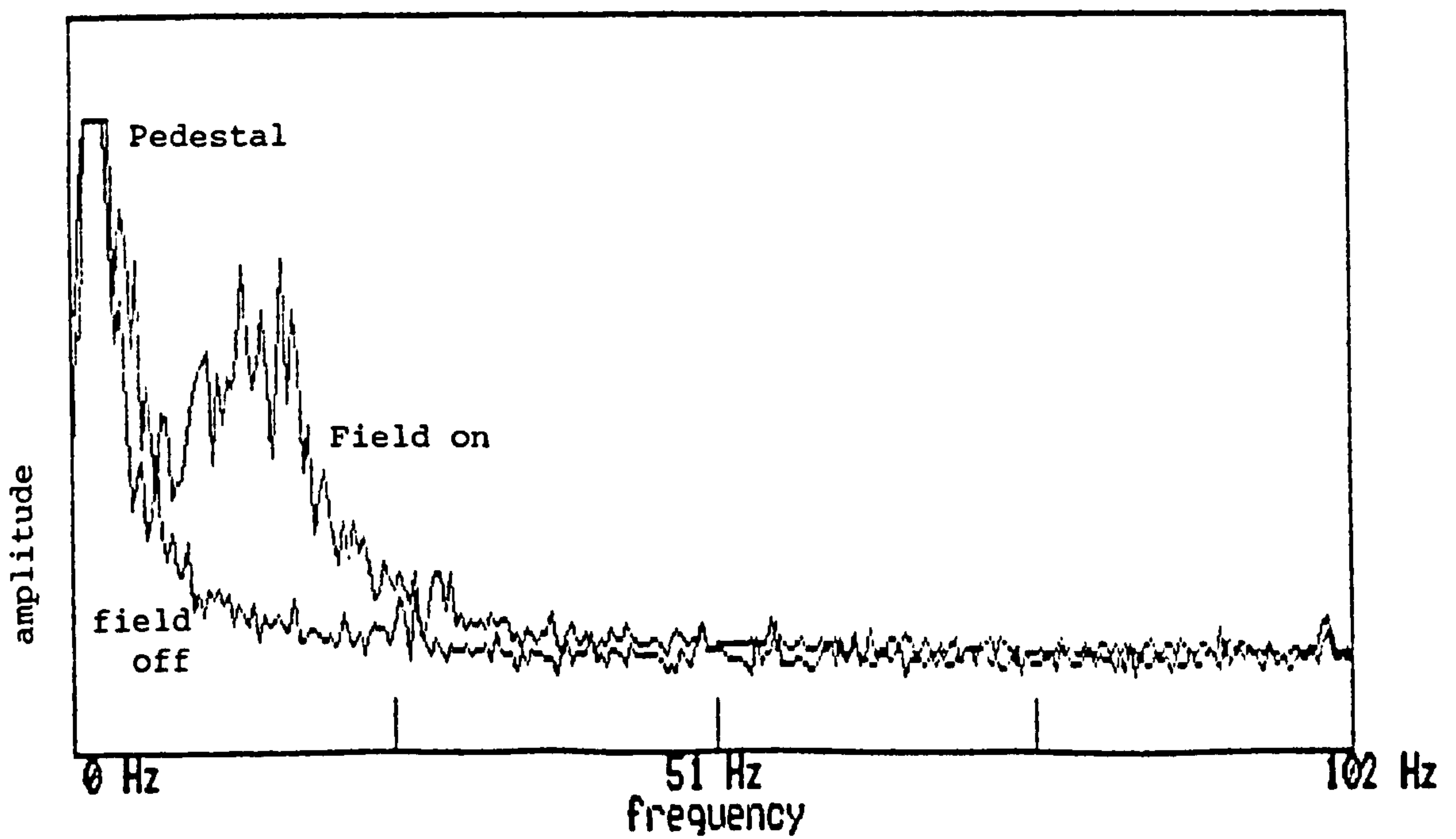


Figure 4.9: Typical static fringe LDE frequency spectrum obtained for an aqueous dispersion.

However, it is the resolving power of the instrument when studying non-aqueous systems that needs to be considered, in particular for non-polar dispersions of small mobility.

#### 4.1.7 NON-AQUEOUS STUDIES WITH THE BASIC VELOCIMETER

The frequency spectra obtained for non-aqueous dispersions using the basic static fringe apparatus were obtained in conjunction with the electrode drive unit originally described by Parkins. He used a custom-built power supply (Mains Interport Supplies), operating in a constant voltage mode. The power supply can generate d.c. voltages between 28V and 670V. The output from the supply unit is connected to a mercury relay switching unit to produce waveforms as shown in figure 4.10. The pulse length,  $T$ , is variable between 80ms and 400ms. The dead time between each pulse,  $D$  ( $= 10\text{ms}$ ), prevents accidental short-circuiting of the power supply. The output from the relay unit provides the signals for the electrodes. A variety of non-polar dispersions were studied using the above electrode drive hardware. Obtaining good quality spectra, and, hence, reliable mobility values, proved to be very difficult. Figures 4.11 (a) to 4.11 (c) are three typical non-polar results. Each figure shows the spectra obtained with and without the electric field applied. Reliable determination of electrophoretic velocities from these spectra is difficult. Many non-polar dispersions yielded spectra which showed little difference when the field was applied. The examples shown

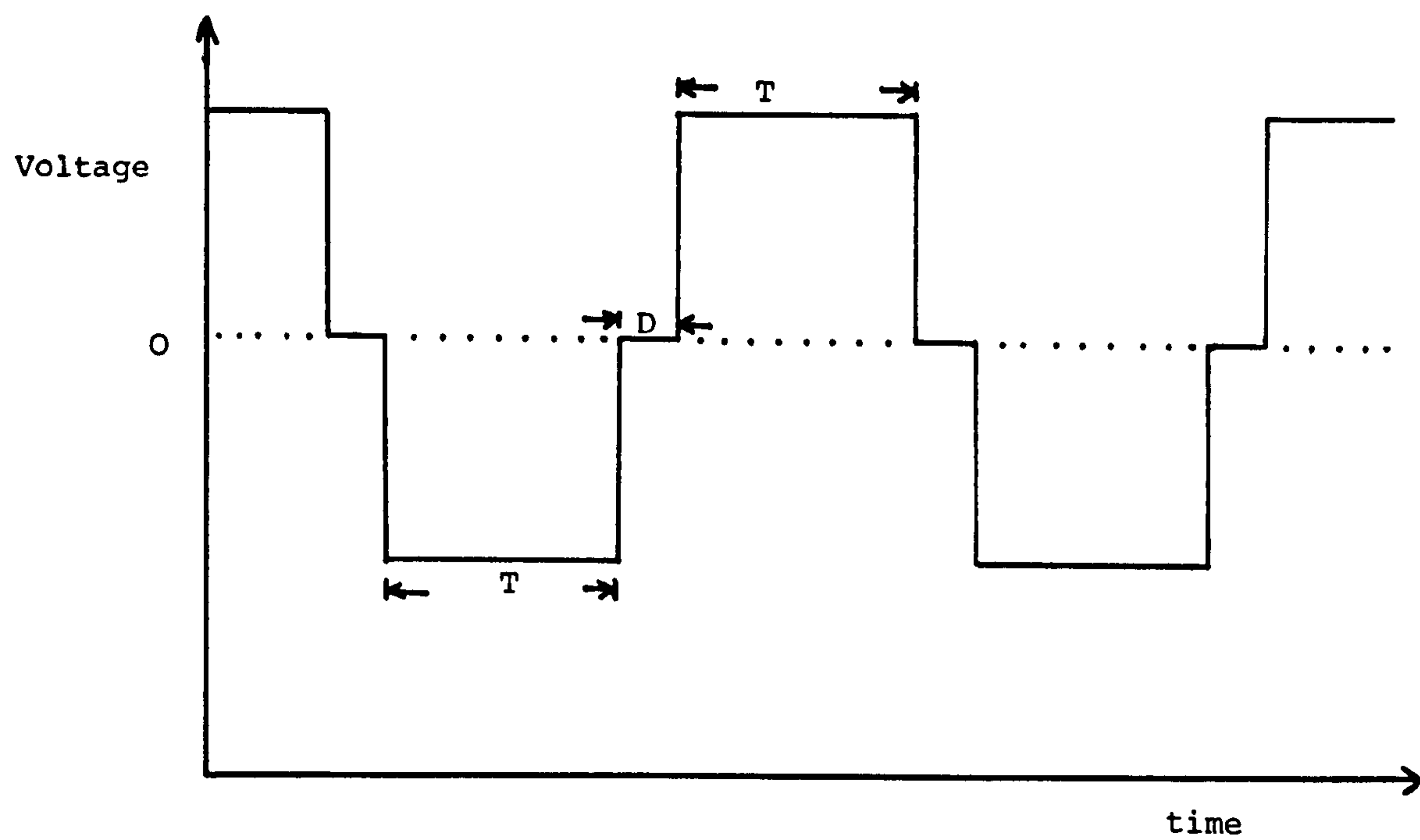


Figure 4.10: Voltage waveform from mercury relay switching unit.

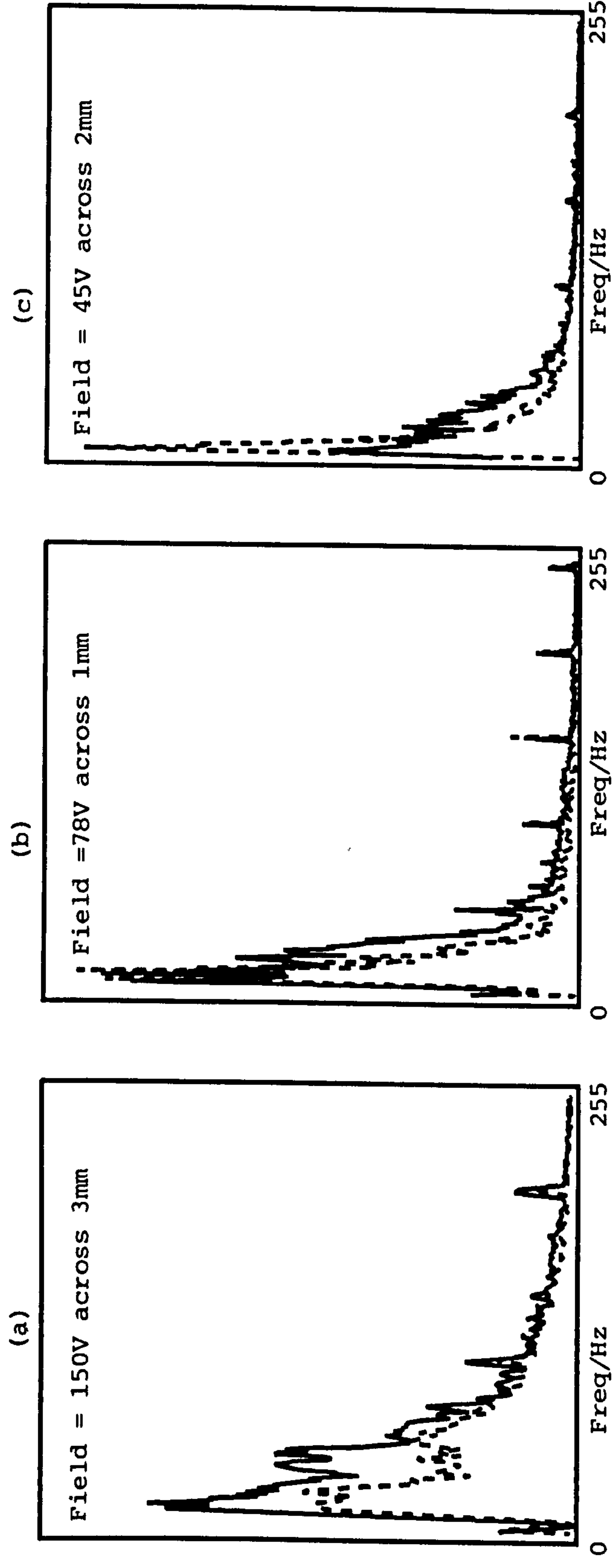


Figure 4.11: Typical frequency spectra obtained for non-polar dispersions using static fringe

velocimeter. Solid line = field on, broken line = field off.

(a) poly(ethylene oxide) -stabilised polypyrrole latex in 1,4-dioxan,

(b) ~ 400 nm diameter stearylated silica particles in cyclohexane,

(c) ~3 $\mu$ m diameter "Hypersil" silica particles in n-hexane/Aerosol OT.

here are for systems with relatively high mobilities ( $\sim 10^{-9} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$ ). Even for these systems, voltages of the order of 100 V need to be applied across the electrodes. Clearly, for non-polar systems with small surface charge densities/potentials, very high, undesirable voltages will need to be employed. Furthermore, with the exception of very large particles ( $\sim 5 \text{ }\mu\text{m}$  diameter), determination of the sign of the electrophoretic velocity is very difficult. Large particles can be seen directly through the PM tube assembly reflex viewer, and their direction of motion determined by eye.

To overcome the serious limitations of the basic static fringe apparatus, it was decided to incorporate the AWPS processing scheme into the equipment. This scheme has been further developed to produce the PALS technique.

## 4.2 DEVELOPMENT OF THE PALS EQUIPMENT

### 4.2.1 OVERVIEW

The development of the PALS equipment from the basic velocimeter design involved three major modifications :

- (i) the generation of moving interference fringes,
- (ii) the determination of amplitude and phase information,
- (iii) processing of amplitude and phase data in order to generate AWPD/AWPS functions etc.



Upon completion of stage (i) above, the apparatus was operated in its spectral analysis mode. The improvement in spectral quality over the static fringe apparatus will be demonstrated later. Determination of the phase and amplitude information using phase-sensitive detectors (described below) was performed using two varieties of detector. The second design is the one from which further developments were made. The software written to process the raw amplitude and phase data evolved over a period of months, with new features being added as the need arose. Complete listings of the final (to date) source code, and associated support files, are given in Appendix 2.

#### 4.2.2 GENERATION OF MOVING INTERFERENCE FRINGES

In § 3.5.5, a method was described of shifting the frequency of laser radiation by using a Bragg cell (AOM). In this section, the hardware used to generate the necessary electrical signals for suitable Bragg cells is described.

The choice of Bragg cell is dictated by the laser wavelength, beam diameter and power, and the frequency modulation bandwidth required. In the latter case, this is simply the frequency difference required between the two beams ( $\sim 1 - 2$  kHz). The Isomet Model 1205C



AOM was chosen for use with the 5 mW He-Ne laser. This AOM has an operating centre frequency of 80 MHz, and, for a 0.4mm diameter He-Ne laser beam, provides a modulation bandwidth at 80 MHz of ~ 5 MHz. The 1205C requires an RF signal of ~ 80 MHz to drive the acoustic transducer, up to 0.6 W in power (+ 28 dBm). The deflection efficiency of the Bragg cell being driven at 0.6 W for the laser in question is quoted as 85% at 80 MHz<sup>109</sup>. Two other features make this AOM a good choice for the PALS equipment. It has good temperature stability and is relatively small (~ 51 x 16 x 22 mm<sup>3</sup>). By incorporating two such Bragg cells into the velocimeter, two RF signals are required of power + 28 dBm and frequencies  $\omega_b$  and  $\omega_b + \Delta\omega_s$ , where  $\omega_b$  is the centre frequency of the Bragg cell (80 MHz) and  $\Delta\omega_s$  is the shift frequency required between the two beams. Figure 4.12 shows a block diagram of the electronics required, in principle, to obtain the necessary Bragg cell signals. The kind of mixer required to generate  $\exp i (\omega_b + \Delta\omega_s)t$  from  $\exp i \omega_b t$  and  $\exp i \Delta\omega_s t$  is a single side-band modulator (SSBM). In choosing an SSBM, it is necessary to realise that thermal fluctuations in the electronics may cause unpredictable frequency drifts of the Bragg cell signals. It was found that commercially available SSBMs could not provide the necessary thermal stability and narrow operating bandwidth without resorting to customised modification of existing units. This was considered to be too expensive. Hence, a custom-built modulator was used (Read Electronics, Clifton, Bristol). This is a mains powered unit with one input and two outputs. The input is the signal from the

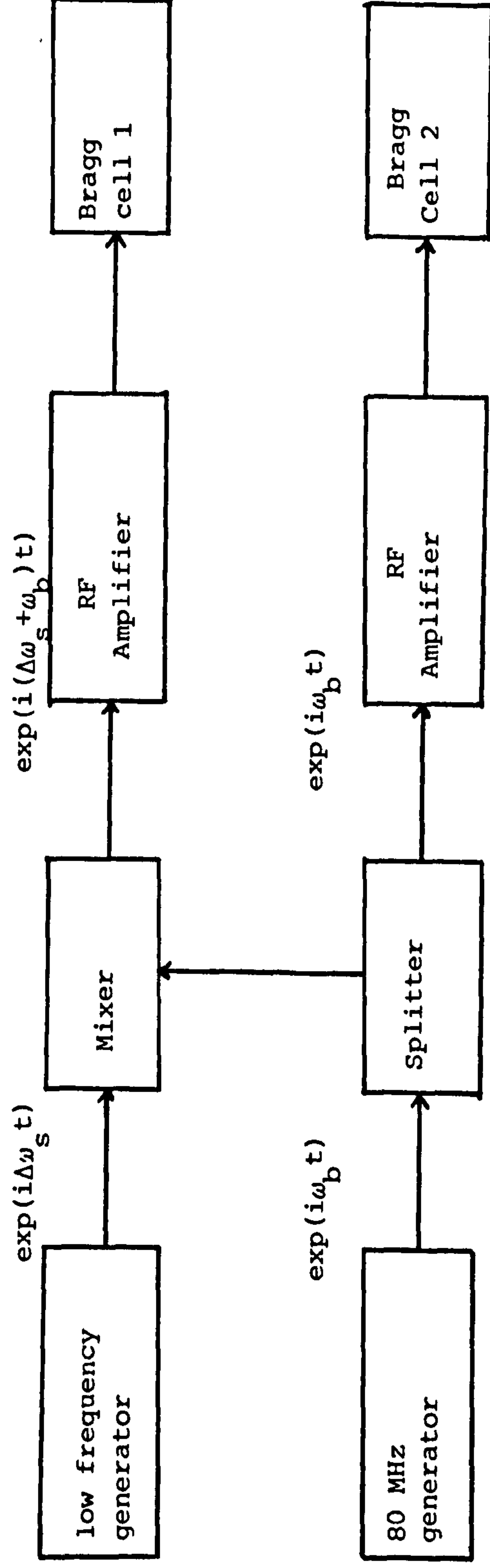


Figure 4.12: Schematic diagram of electronics required for generating Bragg cell signals.

low frequency generator, which will act as the reference signal for the phase demodulation, and the two output signals are 0.6W RF signals. One is 80 MHz and the other is  $80 \text{ MHz} + \Delta\omega_s$ . The unit does not generate the 80 MHz signal directly and then mix it with the low frequency signal. Instead, it generates two intermediate signals, one at 10.7 MHz and the other at 69.3 MHz. The 80 MHz signal is generated by simply mixing the two intermediate signals. The upper side-band signal is obtained by first mixing the 10.7 MHz signal with the low frequency signal and then mixing this with the 69.3 MHz signal. Any thermal fluctuations occurring in either intermediate signal will affect both RF outputs to the same degree. This provides very good frequency difference stability between the two Bragg cell signals. The SSBM unit accepts an input signal with an amplitude between 0.5V and 10V. A variable potentiometer enables the level of the signal to be optimised for the mixing circuitry. The two output signals have the desired power of 0.6W (into a  $50\Omega$  load). The frequency of the input signal can range from 500 Hz to  $\sim 2.5$  kHz. Outside this range, the unit's performance is reduced.

The power of the deflected (i.e. frequency shifted) laser beams depends on the incident beam's power and the amplitude of the Bragg cell drive signal. It is, therefore, very important that the Bragg cell drive signals have constant amplitude. Unfortunately, however, it was found that the RF signals from the SSBM unit are

amplitude-modulated at the frequency of the reference signal. This is a very undesirable situation since the PALS technique involves measuring the phase of a signal relative to the reference signal. The amplitude modulation on the Bragg cell signals causes the laser beams to be amplitude-modulated at  $\Delta\omega_s$ , too. This manifests itself as an unwanted component in the scattered light frequency spectrum, occurring at  $\Delta\omega_s$ . Any phase measurements will thus be underestimated if this interference is too strong. This is similar to the problem described by Schätzel<sup>82</sup>, where crosstalk between the Bragg cell drivers leads to an unwanted signal at  $\Delta\omega_s$ . Since all velocity information obtained for a sample is derived from phase changes of the scattered light signal relative to  $\Delta\omega_s$ , the amplitude-modulation interference must be minimised. This can be achieved by over-amplifying the two Bragg cell signals before they enter the Bragg cells. Two Merrimac GAM - 30 - 150 RF power amplifiers are used. These have a gain of + 30 dB, hence, before amplification, the two Bragg cell signals pass through Merrimac AR-2 20 dB variable attenuators. The amplifiers are powered by a + 20 V d.c. power supply unit built in-house. Unfortunately, the maximum power output from the amplifiers is + 17 dBm, which is somewhat lower than the desired + 28 dBm required for the Bragg cells to operate at maximum efficiency. However, the advantage gained by removing the amplitude modulation interference far outweighs the reduction in beam power arising through the use of + 17dBm signals.



The low frequency signal generator used to provide the reference signal is a Thandar TG102 function generator. This can generate triangle, square or sine waves with amplitudes from 30 mV to 10V over a frequency range of 0.2 Hz to 2 MHz in 6 overlapping decade ranges. Figure 4.13 illustrates the electronic arrangement for generating the required moving interference fringes. A schematic diagram of the optical arrangement for generating the moving fringes has already been given in figures 3.12 and 4.1.

Initial verification of the generation of moving fringes, with a fringe frequency equal to that being generated by the low frequency signal generator, was first performed by placing a piece of frosted glass at the beam intersection point, to act as a stationary scatterer. The scattered light signal from this glass can be observed using the spectrum analyser. It was found that a very sharp frequency component at  $\Delta\omega_s$  was obtained, with some mains interference occurring at lower frequencies. By alternately blocking each of the beams from falling on the frosted glass, it was possible to minimise the amplitude modulation problem by varying the attenuator settings of the AR-2 units and the SSBM input gain control, until the frequency spectrum for the scattered light from a single beam showed little or no  $\Delta\omega_s$ .

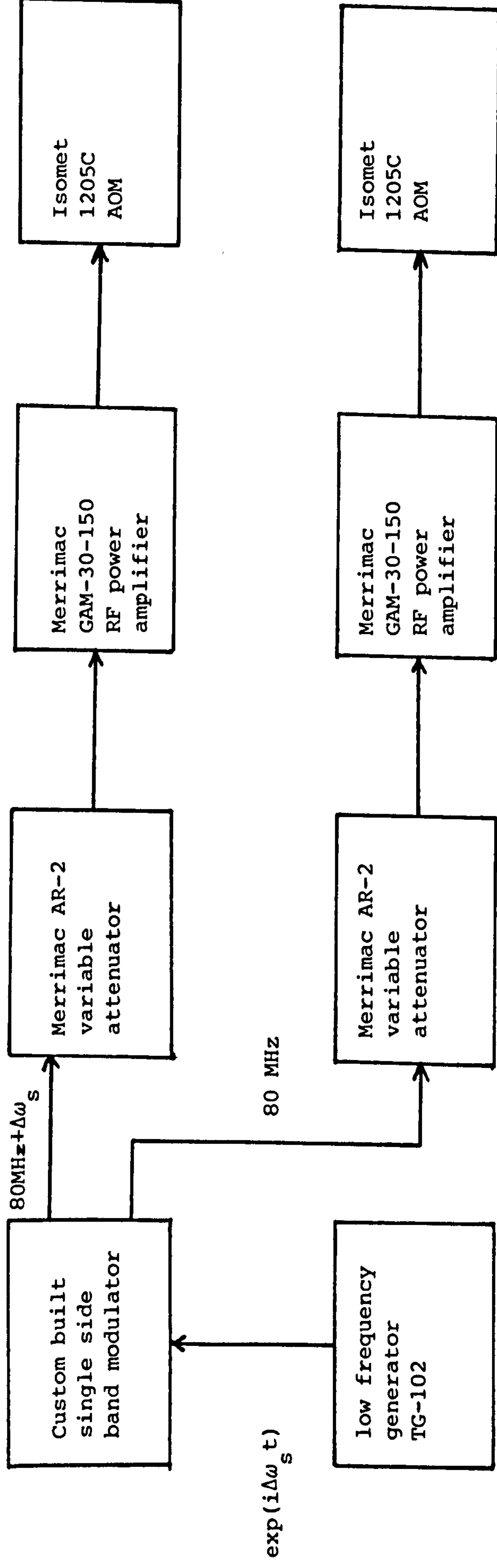


Figure 4.13: Schematic of hardware used to generate moving interference fringes.



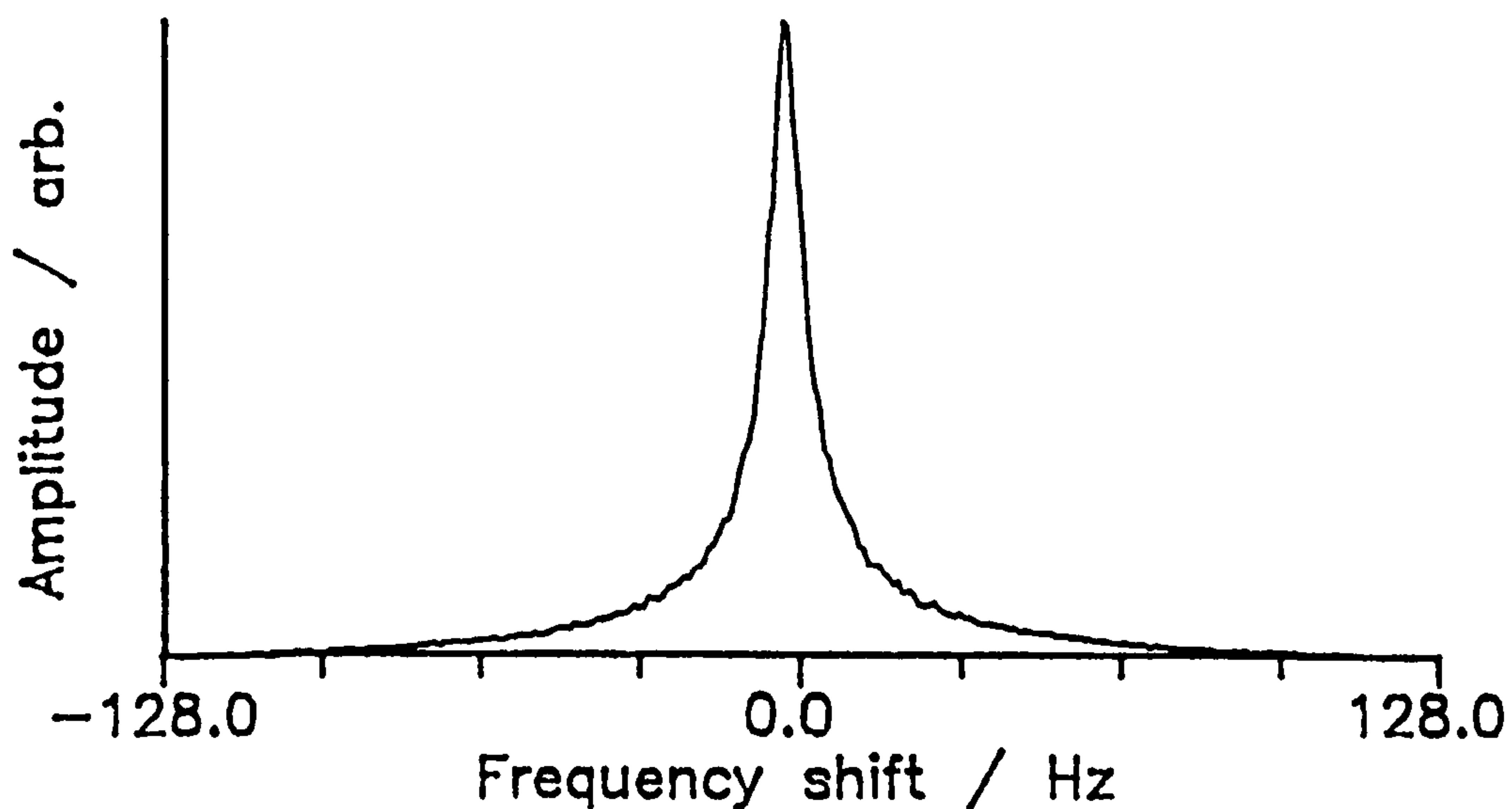
component. It was discovered later that even with the laser and PM tube turned off, a sharp peak at  $\Delta\omega_s$  was sometimes observable. However, when the PM tube anode was disconnected from the filter/amplifier, the  $\Delta\omega_s$  component completely disappeared; i.e. the PM tube assembly itself can act as a radio receiver, picking up stray signals from the Bragg cell signal RF circuitry and heterodyning them to produce a weak signal with frequency  $\Delta\omega_s$ . This problem was alleviated by ensuring that the screened co-axial cable connecting the SSBM to the attenuators was routed away from the PM tube assembly.

All experiments performed using moving fringes were performed at a scattering angle of  $15.4^\circ$  in air, unless stated otherwise.

#### 4.2.3 CONVENTIONAL PROCESSING USING THE MOVING FRINGE VELOCIMETER

As has been explained in Chapter 3, the incorporation of moving interference fringes into a basic velocimeter enables the pedestal signal to be removed and the sign of the observed velocities to be resolved. Figure 4.14 shows the moving fringe frequency spectra obtained for a dilute aqueous dispersion of  $\sim 2 \mu\text{m}$  diameter polystyrene latex

(a)  
 FREQUENCY SPECTRUM FOR PSLg(-)6 IN WATER  
 FIELD: OFF



(b)  
 FREQUENCY SPECTRUM FOR PSLg(-)6 IN WATER  
 FIELD: 2V / 1.56 mm at 2.51 Hz (square)

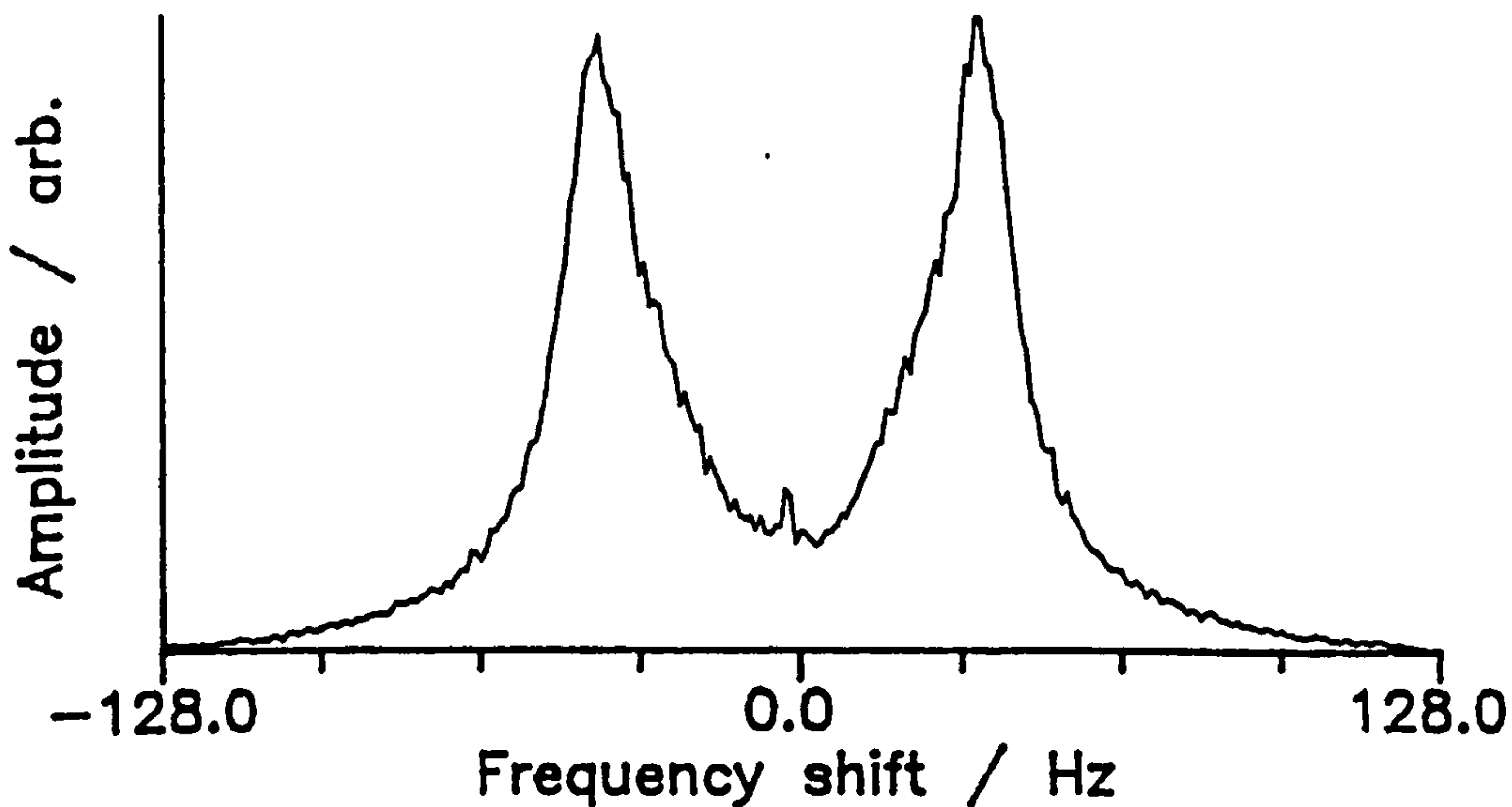


Figure 4.14: Moving fringe LDE spectra for  $\sim 2 \mu\text{m}$  diameter polystyrene latex particles dispersed in water:  
 (a) with no field applied,  
 (b) with square wave field applied.

particles<sup>110</sup>, (a) without the field applied and, (b), with the field applied. 0Hz corresponds to the fringe frequency, in this case 1.955kHz. The general characteristics of these spectra are in excellent agreement with the expected spectra as depicted in figure 3.13. It is evident that this particular sample exhibits a slight polydispersity in its electrophoretic mobility, resulting in an increased broadening of the Doppler peaks. The small peak at 0Hz in case (b) is an example of the amplitude modulation problem described earlier. However, this sample was studied before the realisation that the PM tube assembly can act as a radio receiver. Since then, the presence of sharp peaks at the centre frequency only arises when studying samples with very low scattering properties, which requires higher amplification of the PM tube anode signal.

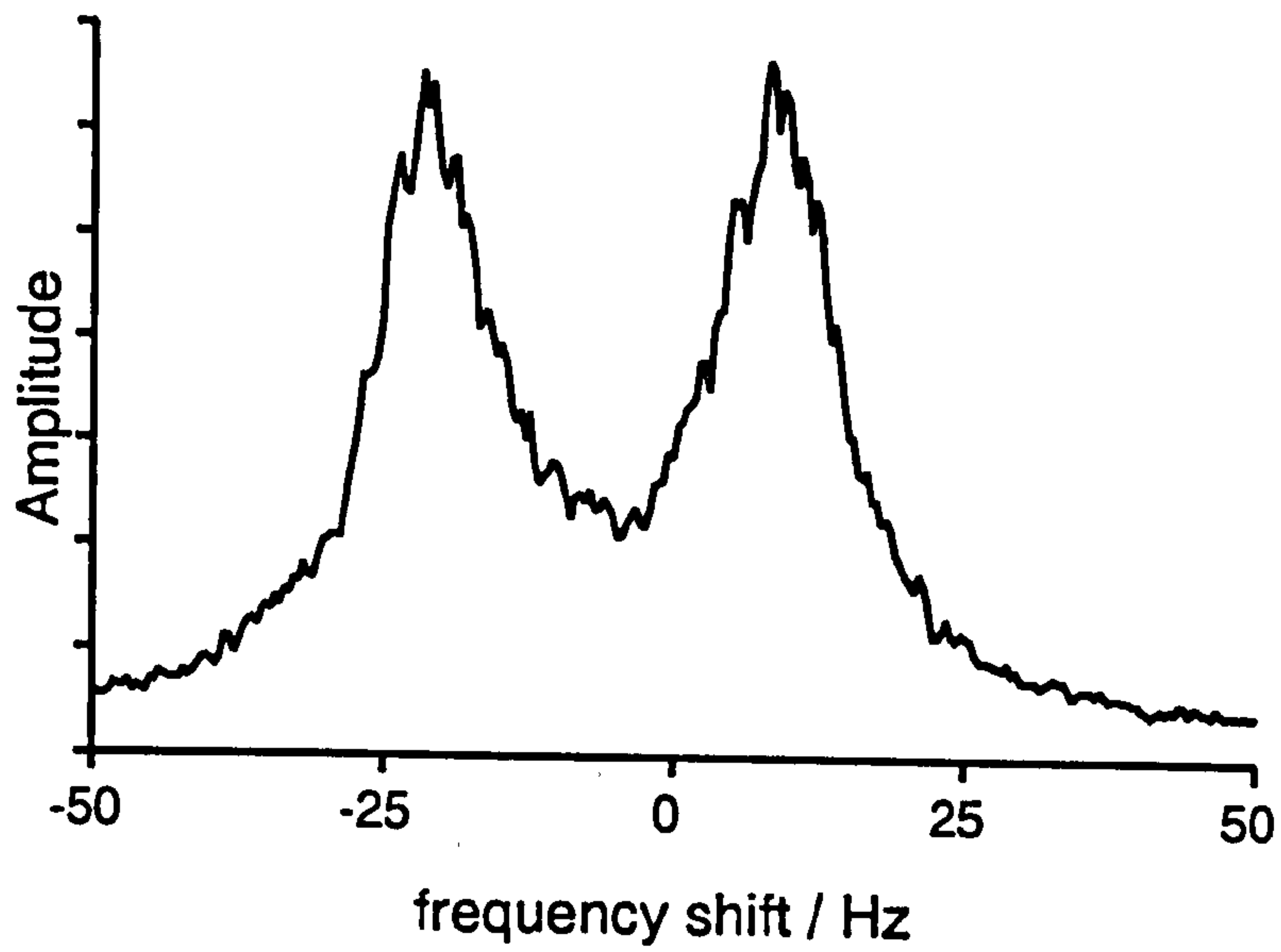
The presence of two electrophoresis peaks in spectrum (b) arises from using a square wave electric field for the reason described previously. The electrode drive signals are generated using a Thandar TG-102 function generator, as used to provide the "reference" frequency. In addition to the general features of this unit described in the previous section, it also provides a variable d.c. offset facility and a synchronous TTL-compatible trigger signal. Either of these two facilities can be used to help determine the sign of the electrophoretic mobility of the sample under investigation. Addition of a d.c. offset to the electrode drive signal will displace the frequency spectrum linearly in a direction dependent on the sign of the mobility of the

sample. The actual direction of displacement of the spectrum depends on the relative geometries of the moving fringes and the electrodes. Alternatively the TTL trigger can be used in conjunction with the HP 3582A spectrum analyser's external trigger facility<sup>111</sup>, such that information is only gathered by the analyser when the electrode signal is positive (or negative). Hence, only one electrophoresis peak will appear in the frequency spectrum. In general, the former d.c. offset option is more convenient, though, perhaps, less versatile.

Figure 4.15 shows frequency spectra obtained for an aqueous dispersion of cross-linked polystyrene latex particles (PSLX-PP-1, see § 5.3 for preparative details) undergoing electrophoresis in a low frequency, square wave field obtained using (a) moving fringes and (b) static fringes. The same sample was used in both cases and the spectra were obtained within 2 minutes of each other. The improvement in resolution and signal to noise ratio by using moving fringes is clear. Figure 4.16 shows the moving fringe spectra obtained for the same particles dispersed in ethanol, both in the absence of an applied field (a), and in the presence of an applied square wave field (b). Hence, it would appear that the moving fringe velocimeter can yield good quality frequency spectra for non-aqueous samples with relatively high electrophoretic mobilities ( $> 10^{-9} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$ ).



(A) MOVING FRINGES (CENTRE = 1955.0 Hz)



(B) STATIC FRINGES

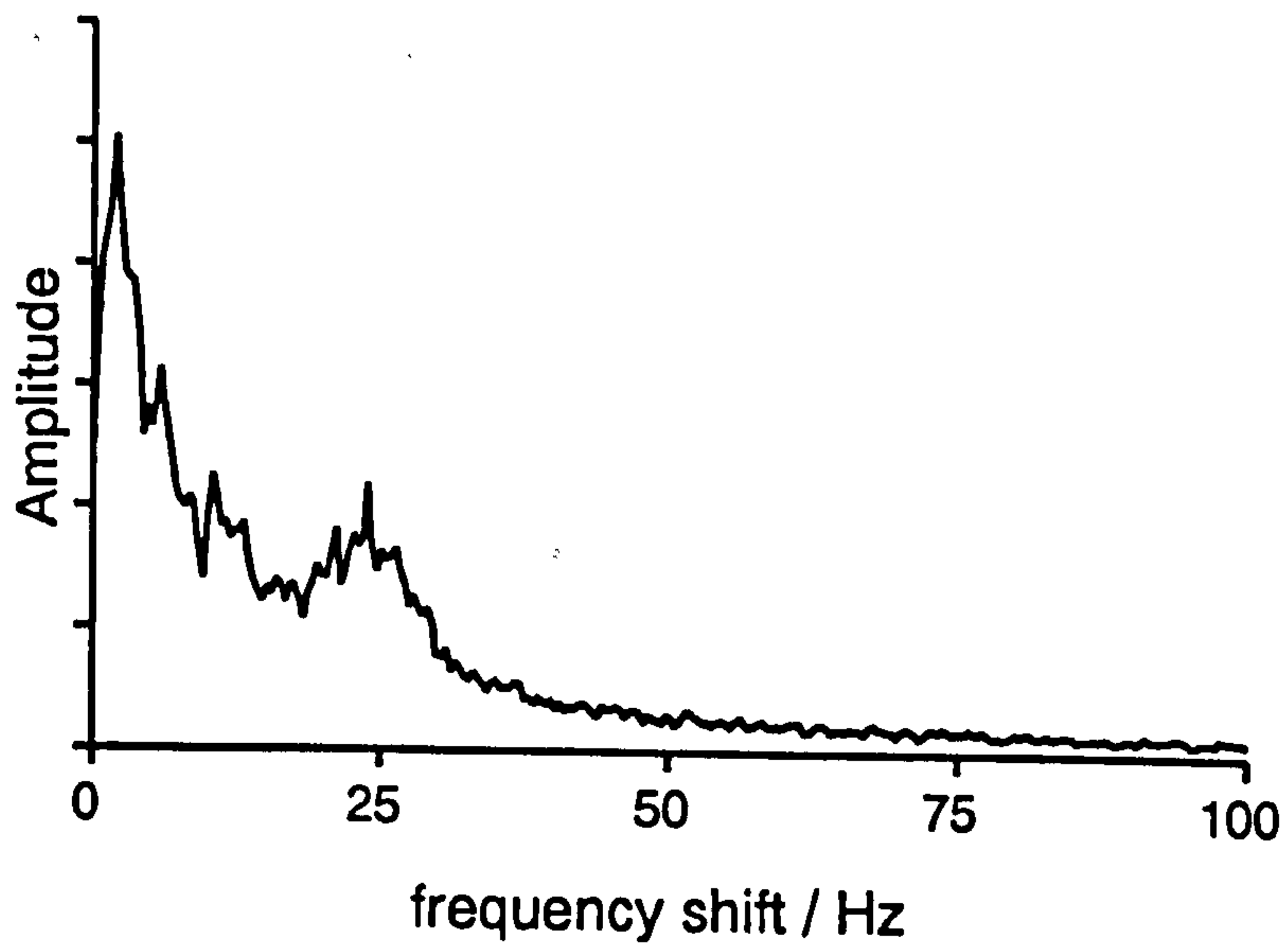
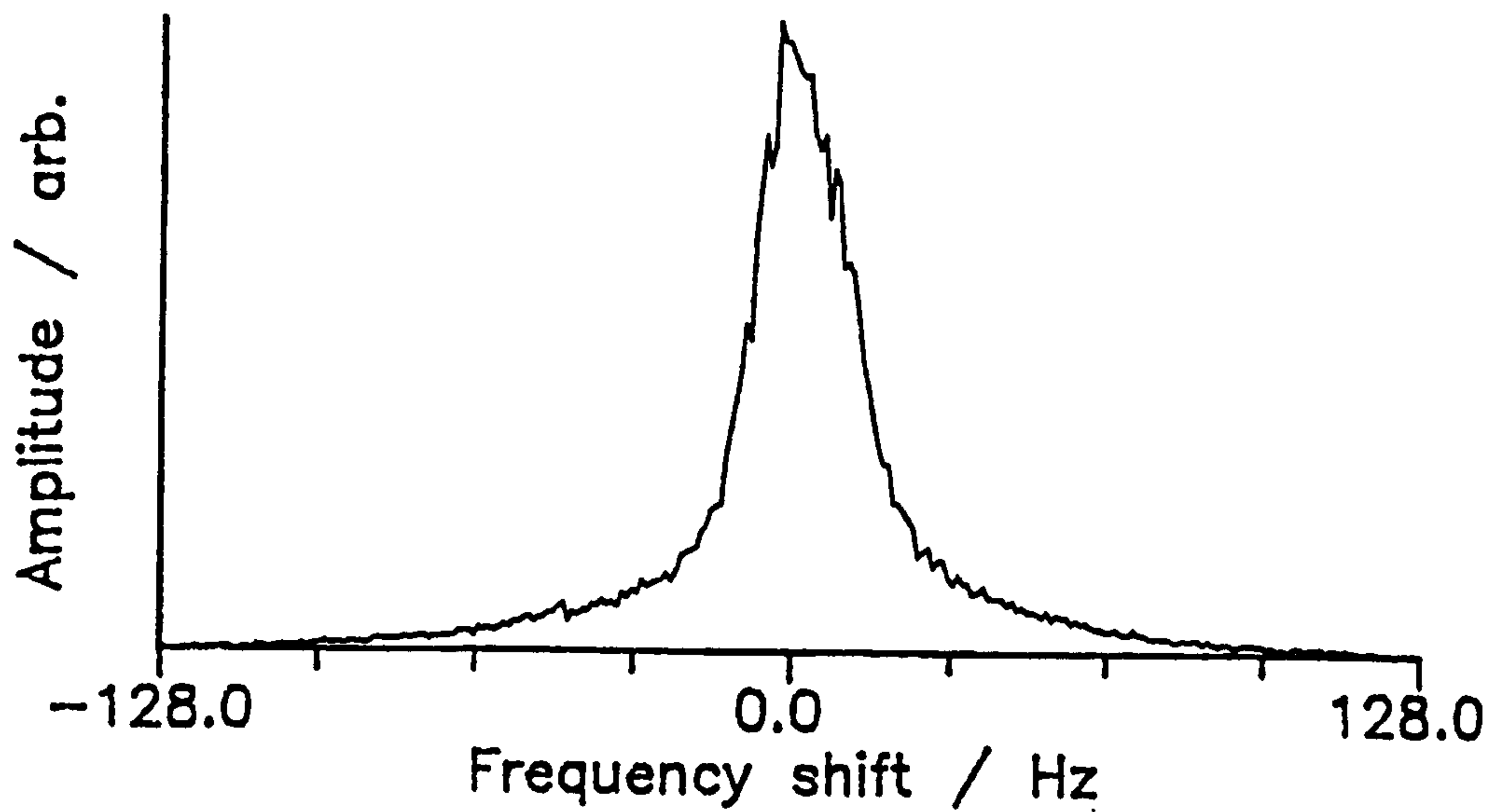


Figure 4.15: Classical LDE spectra for -ve aqueous polystyrene latex dispersion. Electric field =  $7.74 \text{ kV m}^{-1}$  at 3.0 Hz (square wave). (a) shows spectrum obtained using moving fringes and (b) shows spectrum obtained using static fringes.

(a)

FREQUENCY SPECTRUM FOR PSLX-PP-1 IN ETOH  
FIELD: OFF



(b)

FREQUENCY SPECTRUM FOR PSLX-PP-1 IN ETOH  
FIELD: 6V / 1.56 mm at 2.51 Hz (square)  
MOBILITY:  $\pm 6.39 \times 10^{-9} \text{ m}^2/\text{Vs}$

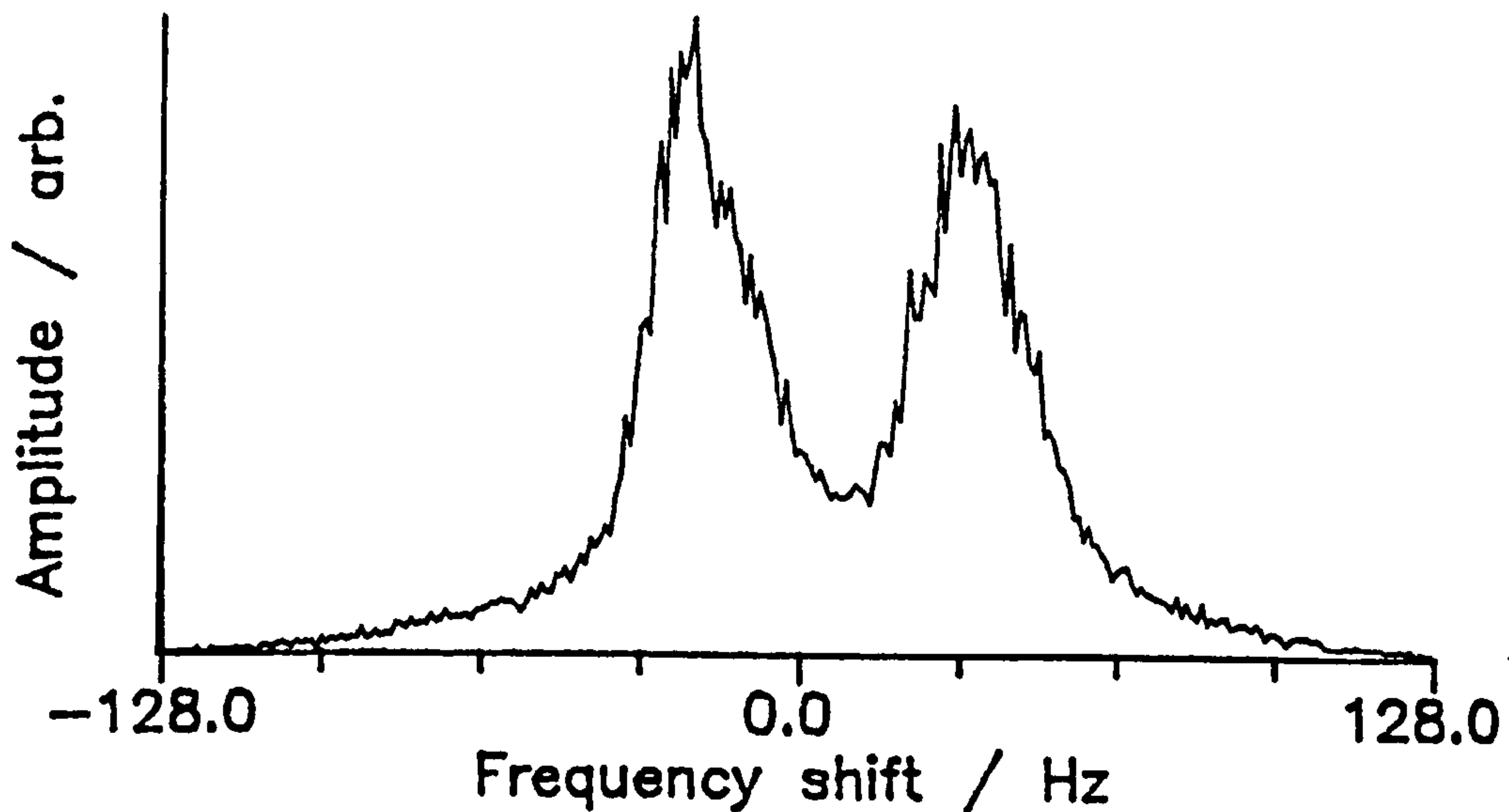


Figure 4.16: Moving fringe LDE spectra for cross-linked polystyrene latex particles dispersed in ethanol:  
(a) with no field applied.  
(b) with square wave field applied.



#### 4.2.4 MEASUREMENT OF AMPLITUDE AND PHASE INFORMATION - THE PRINCIPLES

The determination of the amplitude and phase of the scattered light signal, relative to the reference signal, can be achieved in a number of ways. Schätzel and Merz<sup>102</sup> used a full-wave rectifier and boxcar integrator to determine the amplitude of the band-pass filtered Doppler signal. Integration of the signal over a time approximately equal to one Doppler period provides an estimate of the amplitude. They used a 256 kHz clock generator to create their 1 kHz reference signal. The phase difference between the Doppler signal and the reference is determined by timing the length of a Doppler period, using the 256 kHz clock to drive a suitable counter.

An alternative method provides a direct measurement of the amplitude-weighted phase difference (i.e.  $Q(t) = A(t) \cdot \phi(t)$ )<sup>82</sup>. Consider the two signals represented in figure 4.17. When the filtered Doppler signal crosses zero, a timer is started. When the timer reaches one quarter of the reference period,  $T$ , the Doppler signal is sampled to provide an estimate of the amplitude. After one complete reference period, the Doppler signal is sampled again. At this point, the signal is proportional to  $\sin \phi(t)$ , which, for small  $\phi$ , can be approximated to  $\phi(t)$ . The proportionality constant is the amplitude.

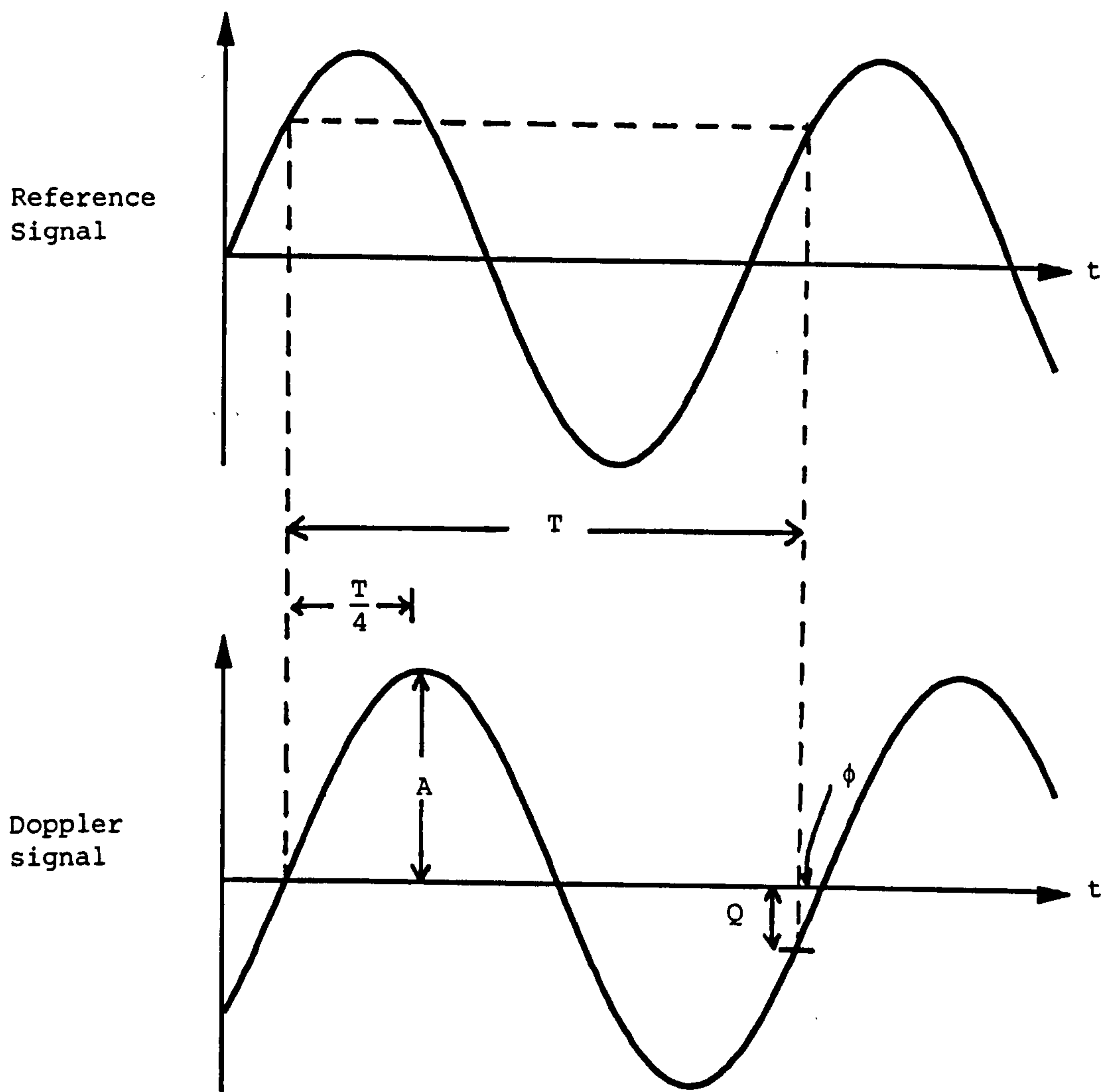


Figure 4.17: Sampling scheme to obtain  $Q(t)$  directly from the Doppler signal (after ref. 82).

Hence, the signal measured one reference period after the initial zero crossing is  $A(t) \cdot \phi(t) = Q(t)$ . This sampling scheme can, in principle, be performed either by hardware or software. However, the latter choice would involve using a computer capable of reading and processing data at a rate of one sample every microsecond if good phase and amplitude resolution are to be obtained for signals with frequencies of the order of 1 kHz.

A more commonly-employed method for the determination of phase is that of phase-sensitive detection. This is the method chosen for use in the apparatus described here, primarily due to the availability of suitable equipment. The basic principles of operation for a phase-sensitive detector (p.s.d.) are as follows<sup>112</sup>. Consider the schematic circuit in figure 4.18. When the reference voltage,  $V_R$ , is positive, the switch (transistor) is closed, allowing the signal current to flow through the load resistor, thereby generating an output voltage,  $V_O$ , proportional to the signal voltage,  $V_S$ . Clearly, if  $V_S$  is in phase with  $V_R$ , then the output voltage will simply be the half-wave rectified form of  $V_S$ . This is shown in figure 4.19. If the signal is  $90^\circ$  out of phase with the reference (i.e.  $V_S$  is in quadrature with  $V_R$ ), the output from the detector will resemble that in figure 4.20. Hence, if the output from the detector is integrated and smoothed, a d.c. voltage will be obtained proportional to the in-phase component of  $V_S$  relative to  $V_R$ . In fact, if the reference voltage is  $V_R \sin \omega_R t$  and the signal voltage is  $V_S \sin (\omega_R t + \phi)$ , the detector output will be a d.c. voltage proportional

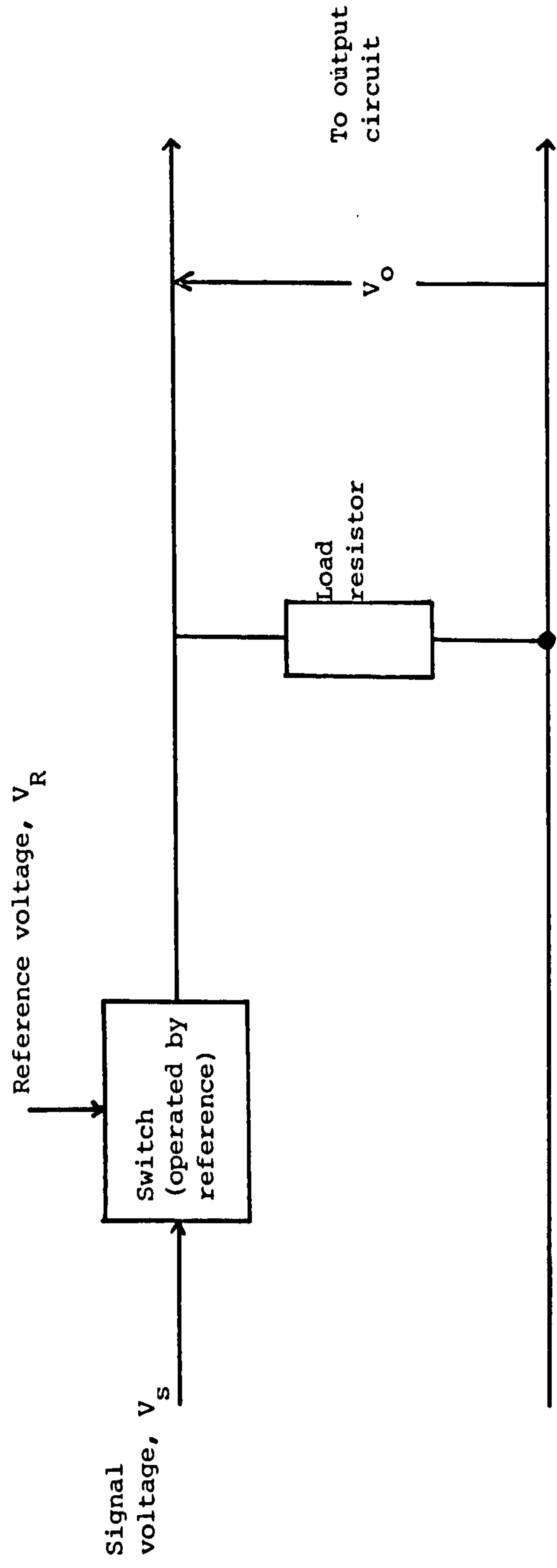


Figure 4.18: Schematic circuit for a phase-sensitive detector.

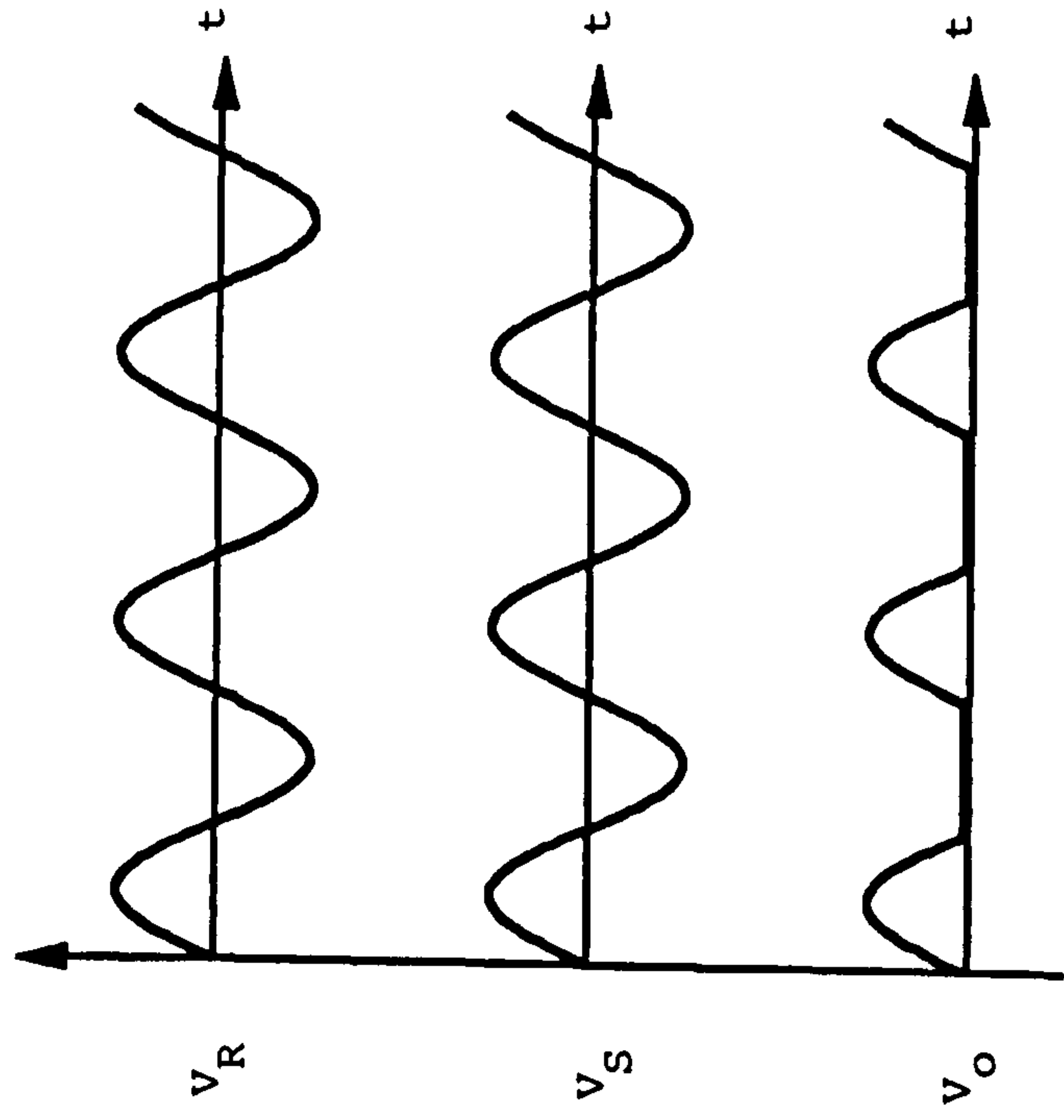


Figure 4.19: Reference ( $V_R$ ), signal ( $V_S$ ) and output ( $V_O$ ) waveforms for a phase-sensitive detector when  $V_R$  and  $V_S$  are in phase.

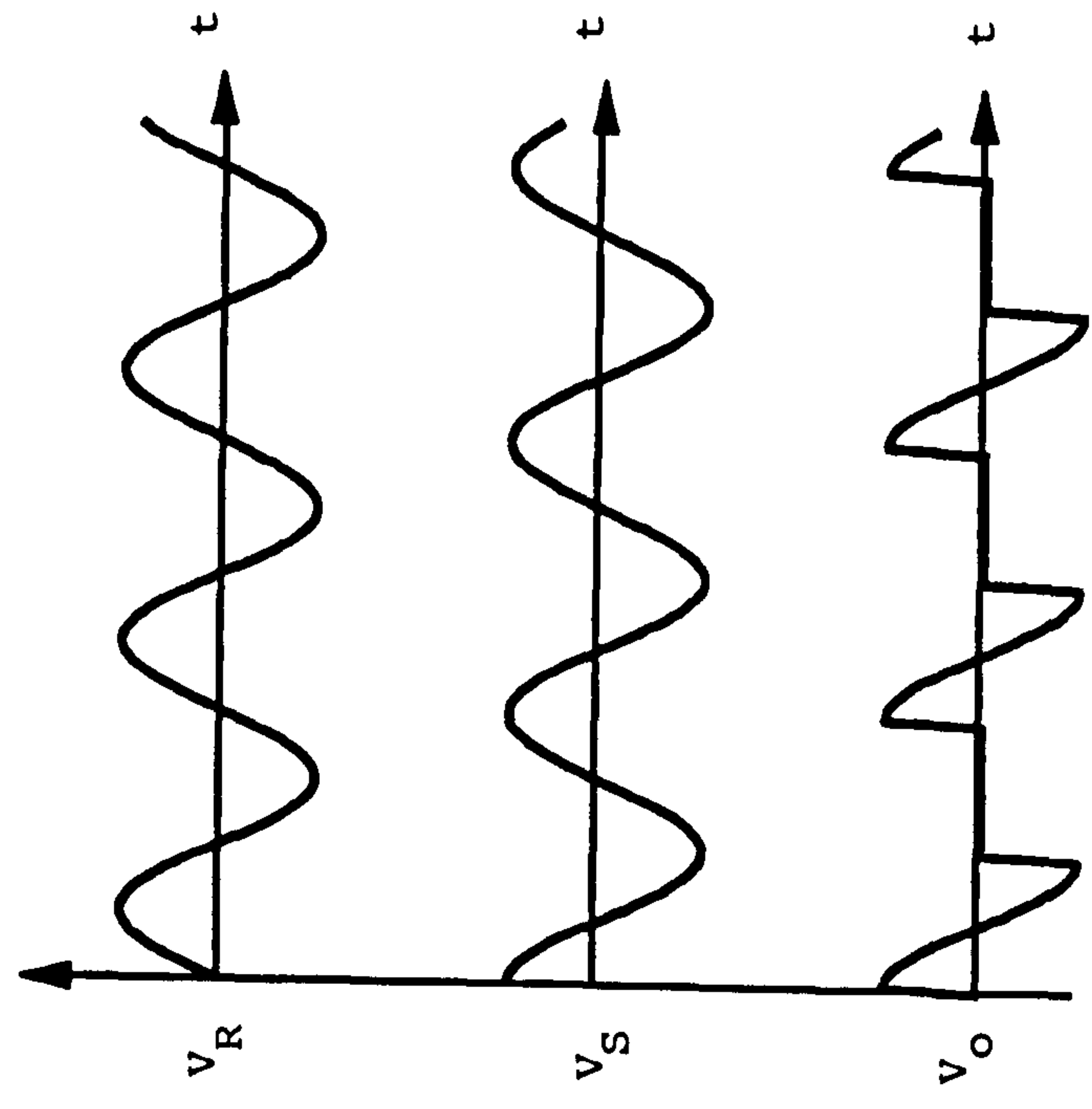


Figure 4.20: As for figure 4.19 except for when  $V_R$  and  $V_S$  are in quadrature.



to  $V_s \cos \phi$ . The operation of the p.s.d. may be thought of as frequency demodulation of  $V_s$  about a centre frequency  $\omega_R$ . Practical designs use a 2-way switch to utilise both positive and negative parts of the signal. A more detailed explanation of such "balanced" p.s.ds can be found in Phillips<sup>113</sup>.

If the reference voltage is phase-shifted by  $90^\circ$ , then the corresponding p.s.d. output will be proportional to  $V_s \sin \phi$ . For the sake of argument, it will be assumed that the gain of the output stage of the detector is such that the proportionality constant for the output voltage is unity. Thus, by using two p.s.ds, driven in quadrature, two signals can be obtained,  $V_s \sin \phi$  and  $V_s \cos \phi$ . Clearly,

$$V_s = \sqrt{(V_s \sin \phi)^2 + (V_s \cos \phi)^2} \quad (4.1)$$

$$\text{and} \quad \phi = \tan^{-1} (V_s \sin \phi / V_s \cos \phi) \quad (4.2)$$

Therefore, both the amplitude and the phase of the input signal, relative to the reference signal, can be readily obtained. Figure 4.21 shows a schematic diagram of how this may be incorporated into the velocimeter design.



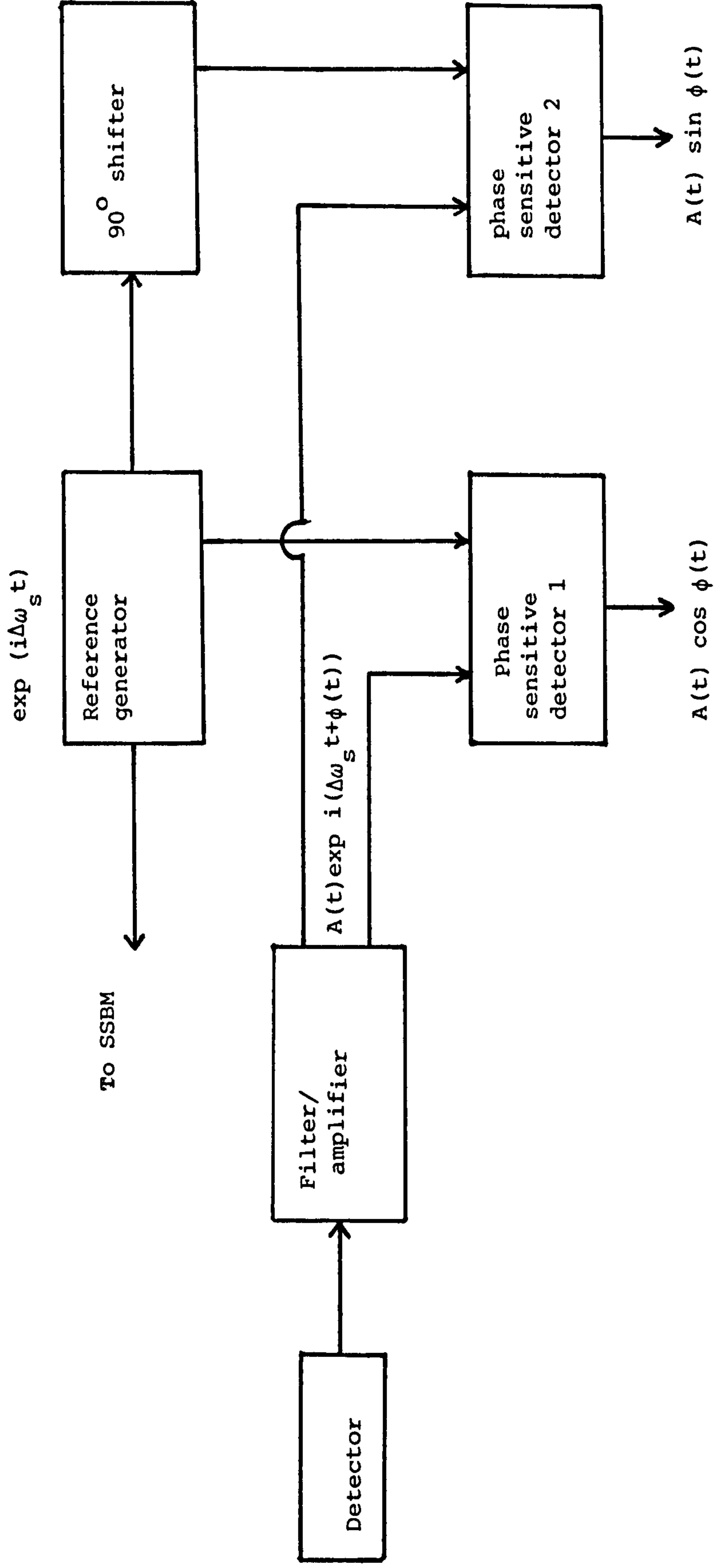


Figure 4.21: Schematic diagram of how to use two phase sensitive detectors to obtain amplitude and phase information for a PALS apparatus.

#### 4.2.5 INITIAL PHASE-SENSITIVE DETECTOR HARDWARE USED FOR PALS APPARATUS

The initial type of p.s.d. incorporated into the velocimeter is the Brookdeal 9412A p.s.d. This model performs the process described above and provides the necessary amplification and post-detector filtering required. Input signal sensitivity can range in amplitude from 1 mV rms to 100 mV rms, to produce maximum output signals of  $\pm 10V$ . The reference signal supplied to the detector can be any symmetric waveform of amplitude 1V to 100V. The operating bandwidth of the 9412A p.s.d. is 0.3 Hz to 3 MHz. It has already been stated that the reference signal generator is a Thandar TG-102 unit, which also has a synchronous TTL output signal. The waveforms shown in figure 4.22 represent the TG-102 outputs when generating sinewaves. The 9412A p.s.d. converts a sinusoidal reference signal into a squarewave. Thus, the TG-102 generator can provide two reference signals in quadrature, each driving a 9412A p.s.d. The Barr and Stroud EF5 filter/amplifier unit is used to amplify and band-pass filter the Doppler signal before being processed by the p.s.d. units. The amplification provided by the EF5 unit and the p.s.ds is optimised to produce output signals with voltages typically in the range of 3 - 10V. Figure 4.23 shows a schematic block diagram of the equipment used for obtaining the amplitude and phase information of the Doppler signal. The means of inputting the p.s.d. outputs into the computer are discussed in the next section.

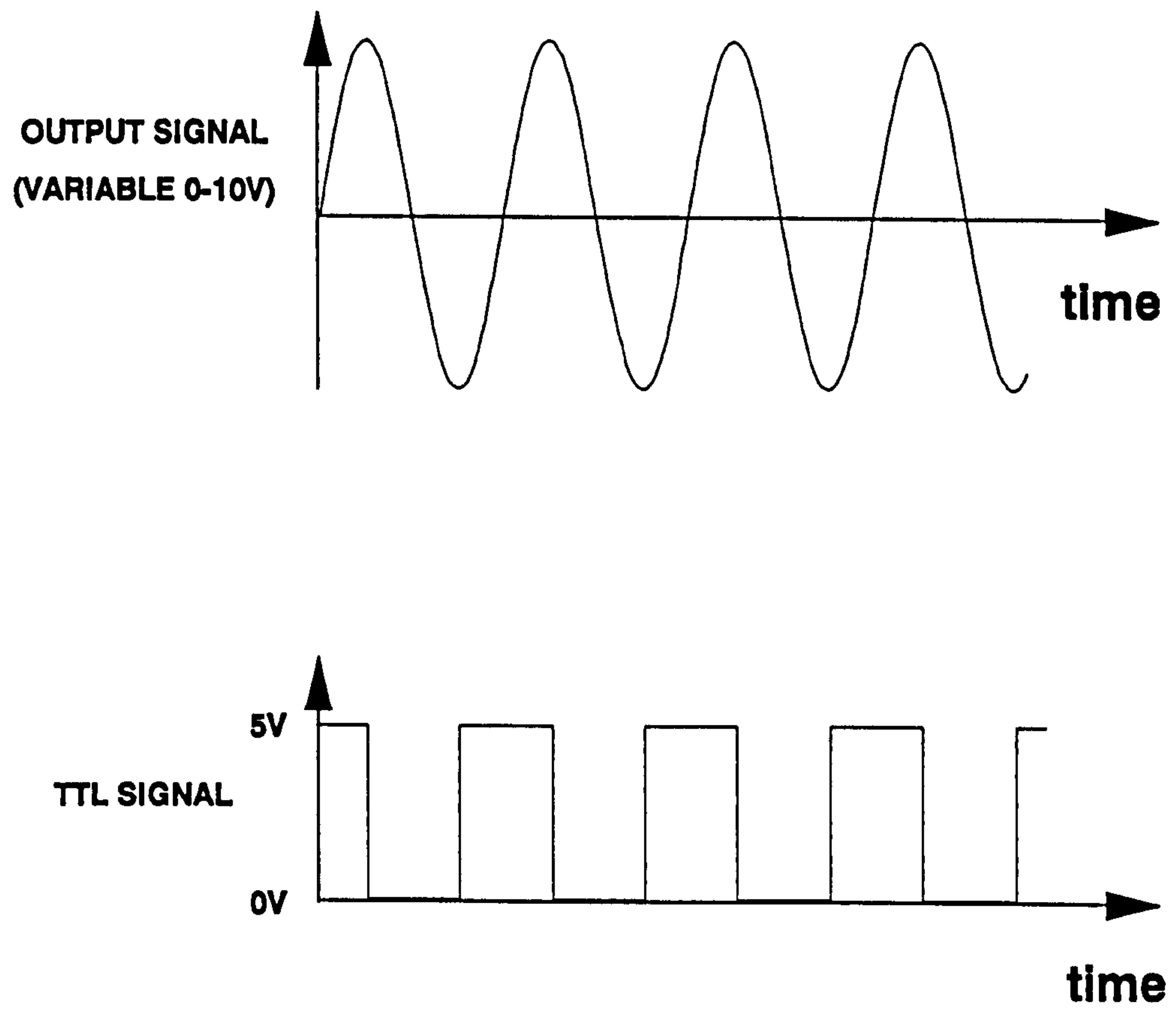


Figure 4.22: Relationship between TG-102 output signal and synchronous TTL signal.

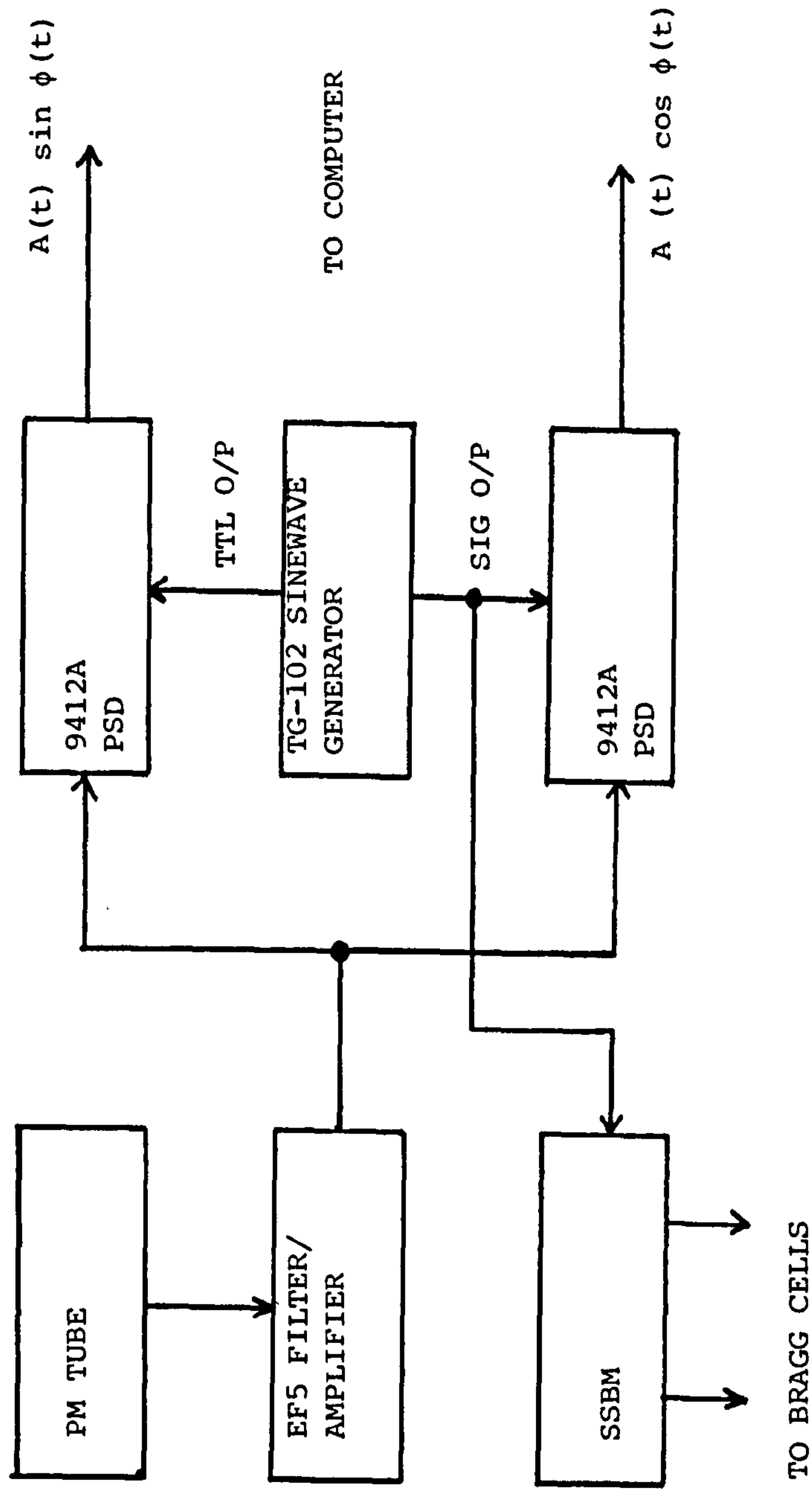


FIGURE 4.23: Schematic representation of detector electronics used in initial PALS apparatus.

#### 4.2.6 ANALOGUE-TO-DIGITAL CONVERTER REQUIREMENTS

In order to process the two p.s.d. output signals from the basic PALS hardware, an analogue-to-digital converter (ADC) is required, which produces digital representations in a computer of the two analogue p.s.d. signals. The ADC should possess at least two independent analogue-to-digital (A/D) channels, capable of reading the signals from both p.s.d.s simultaneously. It should have an on-board programmable clock, such that data can be input at a known, fixed rate. The ADC should be capable of reading in data at a rate equal to the reference frequency. For a frequency of 2 kHz, this requires A/D conversions occurring in less than 500  $\mu$ s. A PALS experiment will be performed such that batches of, say, 50 data point sets are input consecutively into the computer's memory for processing. For a typical sampling rate of  $1 \text{ ms}^{-1}$ , this involves transferring 2000 numbers per second from the ADC to the main computer's memory. Although this throughput rate could be achieved by using a suitable piece of software, it means that the main processor is dedicated to data capture. Processing of the raw data would need to be performed at the end of the experiment or between each batch of data being input. The latter case would mean that the data processed would not be continuous, and both schemes would add an undesirable processing overhead to the duration of the experiment. Fortunately, the HP Vectra



PC, in common with most IBM PC-compatible computers, possesses a direct memory access (DMA) controller. DMA allows a peripheral device, such as a hard disc, to access the computer's main memory whilst the central processor is busy. In effect, it means that the peripheral device and the processor can use the same memory simultaneously without bus management problems occurring. An ADC with a DMA facility would allow raw data to be placed in memory whilst the main computer processes a previous batch of data. By writing suitable software, real-time processing of the amplitude and phase data could be achieved. The means of realising this is described later. One final requirement of the ADC is that it should have an externally-driven trigger facility; i.e. transfer of a batch of data should commence at a point in time determined by, in the PALS case, a fixed point on the electrode cycle. It should also be possible, when required, to override the external trigger, to allow asynchronous/continuous processing of the raw data. The reasons for this have been explained in chapter 3.

The ADC board chosen for use in this work is a Data Translation DT2818 Simultaneous Sample and Hold Single Board Analogue and Digital Input/Output System. The DT2818 provides 4 A/D channels, 2 D/A channels and 16 digital I/O lines. Voltages presented to the A/D inputs are sampled within a 500 ns window and then converted to 12-bit digital numbers. The maximum throughput for A/D conversions is 27,500 samples per second, thereby allowing sampling rates of



13.75 kHz for two channels or 6.875 kHz for all four channels. The input voltages can range from - 10V to + 9.9952 V, yielding 12-bit binary numbers in the range 0 to 4095. Hence, the DT2818 ADC board can measure input voltages to an accuracy of 4.88 mV (0.024% full scale). The on-board clock is programmable in multiples of 1.25  $\mu$ s. An external clock may be supplied, and the trigger used to commence conversions may be internally or externally generated. The digital-to-analogue channels provide -10 V to + 9.9952 V outputs with 12-bit resolution. The digital I/O lines provide a means of sensing or controlling TTL-level compatible control lines from other pieces of hardware. For further, more detailed information about the DT2818, such as programming, the DT2818 User Manual should be consulted<sup>114</sup>.

Physical connections between the external hardware and the DT2818 board are made by use of a Data Translation DT707 Screw Terminal Panel, connected to the DT2818 by a 1-metre 50-way ribbon cable.

#### 4.2.7 SELECTION OF PROGRAMMING LANGUAGE AND GENERAL SOFTWARE OVERVIEW

Borland's Turbo Pascal Version 4.0 compiler was chosen as the programming language for development of the PALS software. This is a high-performance compiler, offering many useful features, including

the ability to access the host computer's low-level input/output ports and disc operating system (DOS). The former is necessary since this allows the DMA controller and the ADC board to be programmed using Pascal. The access to DOS facilities greatly aids the development of the data storage/retrieval aspects of the software. Turbo Pascal's "unit" concept<sup>115</sup> allows low-level routines to be developed and stored in "units", which behave as command libraries. Other programs can then access the commands defined in the units at a higher, more easily-understood level (from the programmer's point-of-view).

For example, all the necessary routines to program the DT2818 A/D board are defined in a unit called "DT2818" and can be accessed by any Pascal program. The software developed for the PALS equipment can be visualised as existing in a number of levels. At the lowest level are the raw, low-level routines. Above that are some high-level routines, which use the low-level routines, and perform tasks such as inputting data from the ADC into memory, via the DMA controller. In the next level are the procedures for performing a PALS experiment, such as initial configuration of the ADC, determination of experimental parameters, data acquisition and data processing. Finally, the highest level can be considered as the user interface. This provides a menu-driven, user-friendly means of performing PALS experiments, with a variety of ways of storing/outputting the results. The user does not require any knowledge of

how the ADC or DMA programming functions.

The other facilities provided by the compiler make it a good choice for this work. Firstly, it provides versatile, high-level procedures for generating graphical displays. Secondly, the compiler supports floating-point maths co-processors. Most microcomputers perform floating-point arithmetic using low-level software routines, which are comparatively slow. IBM PCs (and compatibles) can use a co-processor to perform floating-point arithmetic using hardware, which is much faster. The computer used for this work has a 80387 floating-point maths chip.

The first version of the PALS software performed all its mathematical operations using explicit floating-point arithmetic. A detailed description of this temporary version of the software at this point would be superfluous. A description of the final version of the software accompanies the source code listing, given in Appendix 2.

#### 4.2.8 PRELIMINARY SOFTWARE TESTING

Testing of the data acquisition, processing and function fitting routines was achieved by using a sinusoidal waveform, from a TG-102 generator, to represent the scattered light signal. The radial frequencies of this signal,  $\omega_s$ , and the reference signal,  $\omega_R$ , were determined using the spectrum analyser. Construction of the



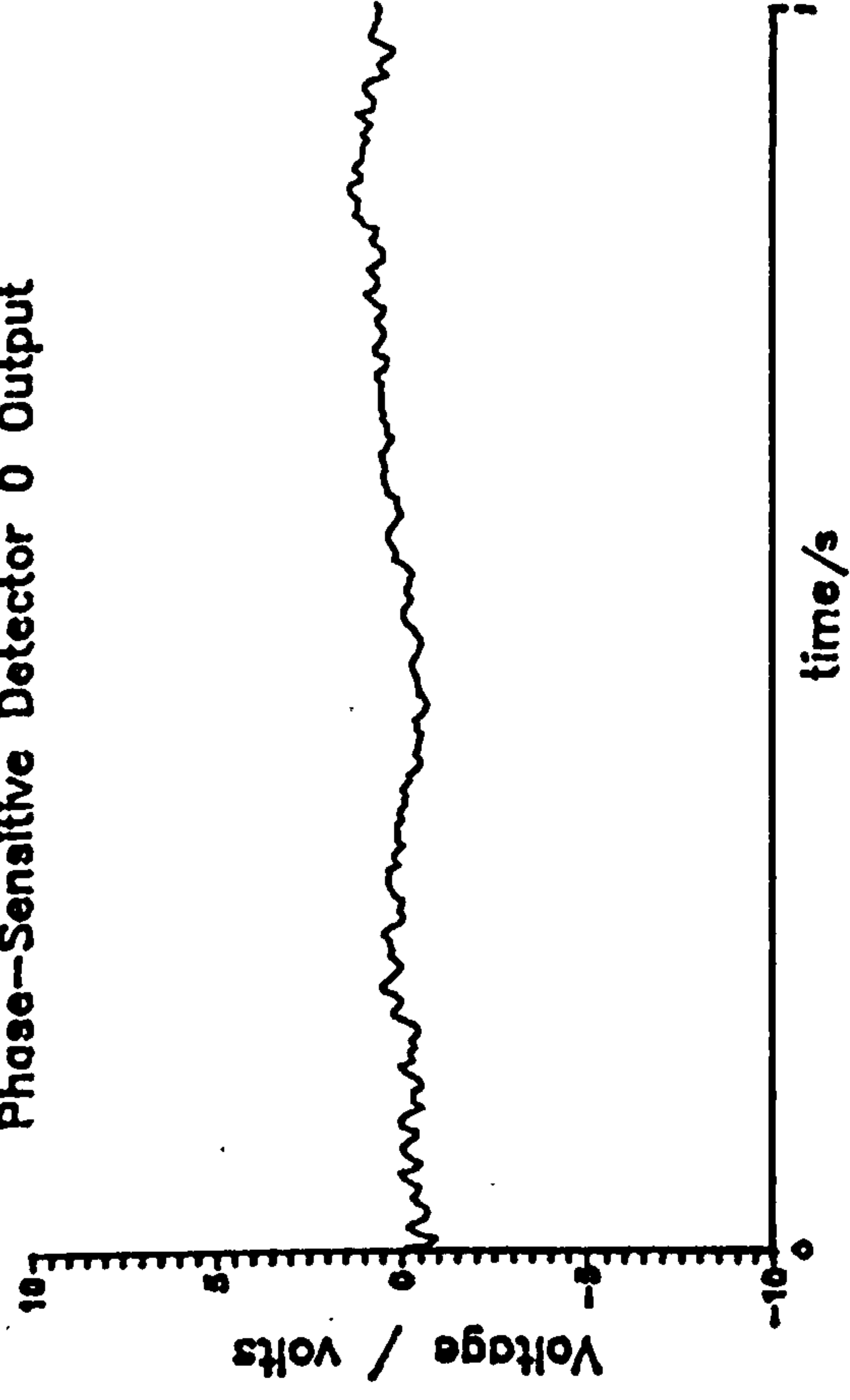
AWPD function should yield a straight line, passing through the origin, with gradient  $\omega_s - \omega_R$  rad s<sup>-1</sup> (§ 3.6.3). This was found to be the case. The fitting procedure used for all functions determined from a PALS experiment is a least-squares approximation method, employing Gaussian elimination with partial pivoting. The Pascal source code is provided by Borland as part of their "Numerical Toolbox". This method can be used for fitting data to any linear combination of variables.

#### 4.2.9 INITIAL INSTRUMENT VALIDATION

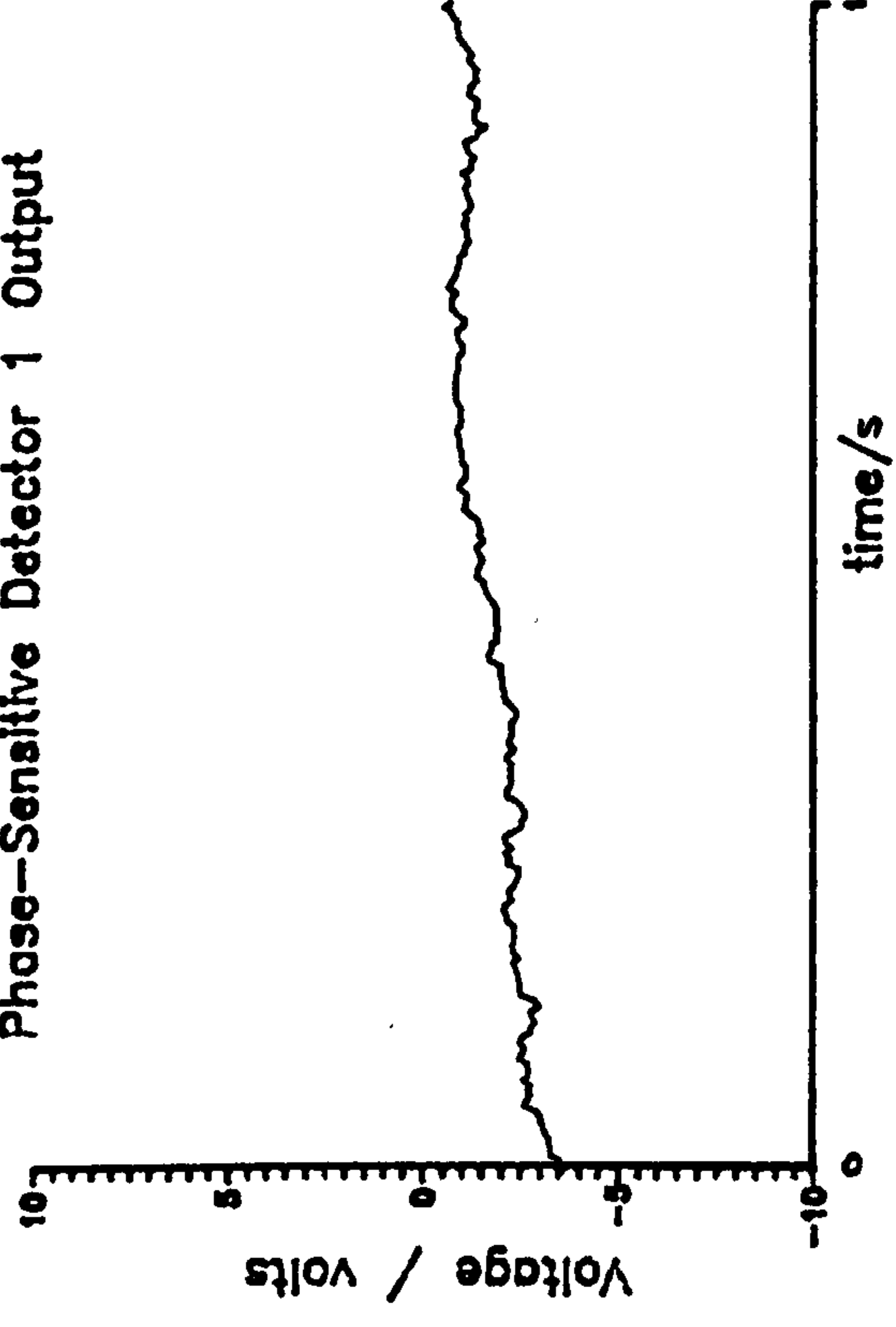
A dilute dispersion of ~ 200 nm diameter zeolite particles in ethylene glycol (ex. Unilever plc) was used to confirm that the PALS hardware yielded the expected amplitude and phase information for a sample undergoing motion in the absence of an electric field and in the presence of a sinusoidal and square-wave field .

Figures 4.24 (a) - 4.24 (c) show the p.s.d. outputs and corresponding amplitude and phase information for the zeolite dispersion (a) with no field applied, (b) with a squarewave field applied and (c) with a sinusoidal field applied. In each case, the y-axis scale for the phase data represents the equivalent of one fringe spacing. For case (a), the phase data shows that both high frequency mains interference and slight diffusive contributions are present. Case (b) shows that the response of the particles to a squarewave field changing polarity

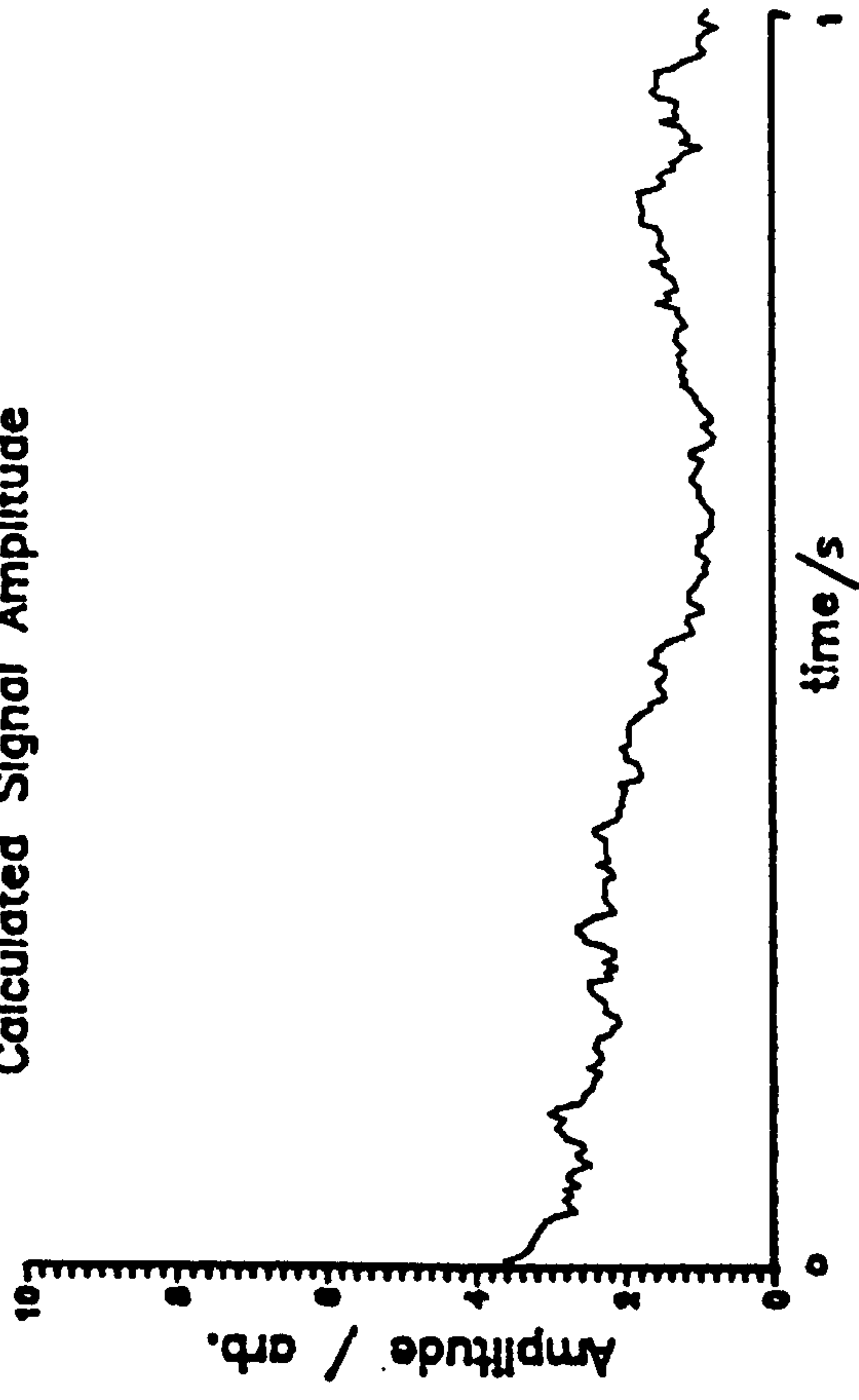
Phase--Sensitive Detector 0 Output



Phase--Sensitive Detector 1 Output



Calculated Signal Amplitude



Calculated Signal Phase

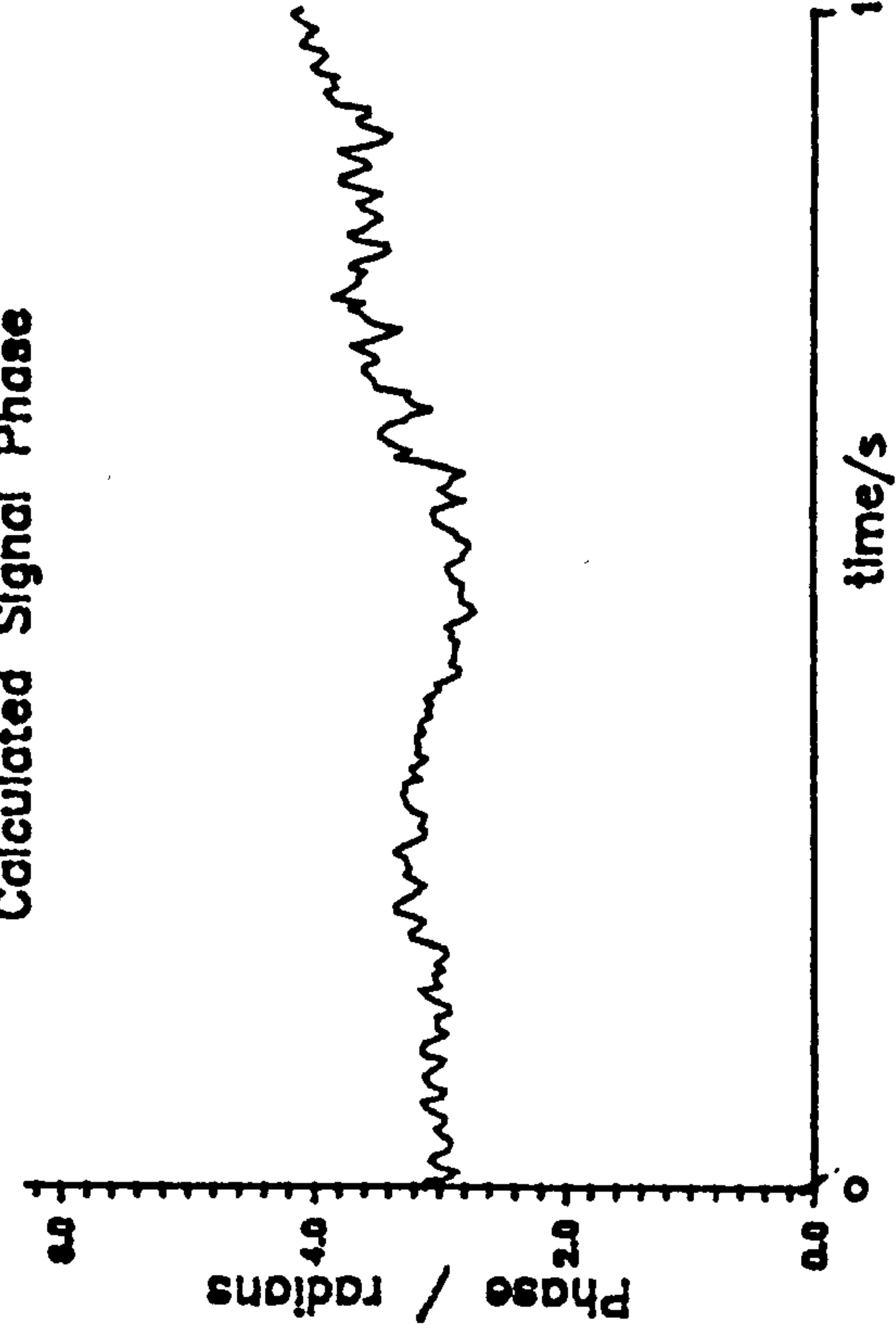
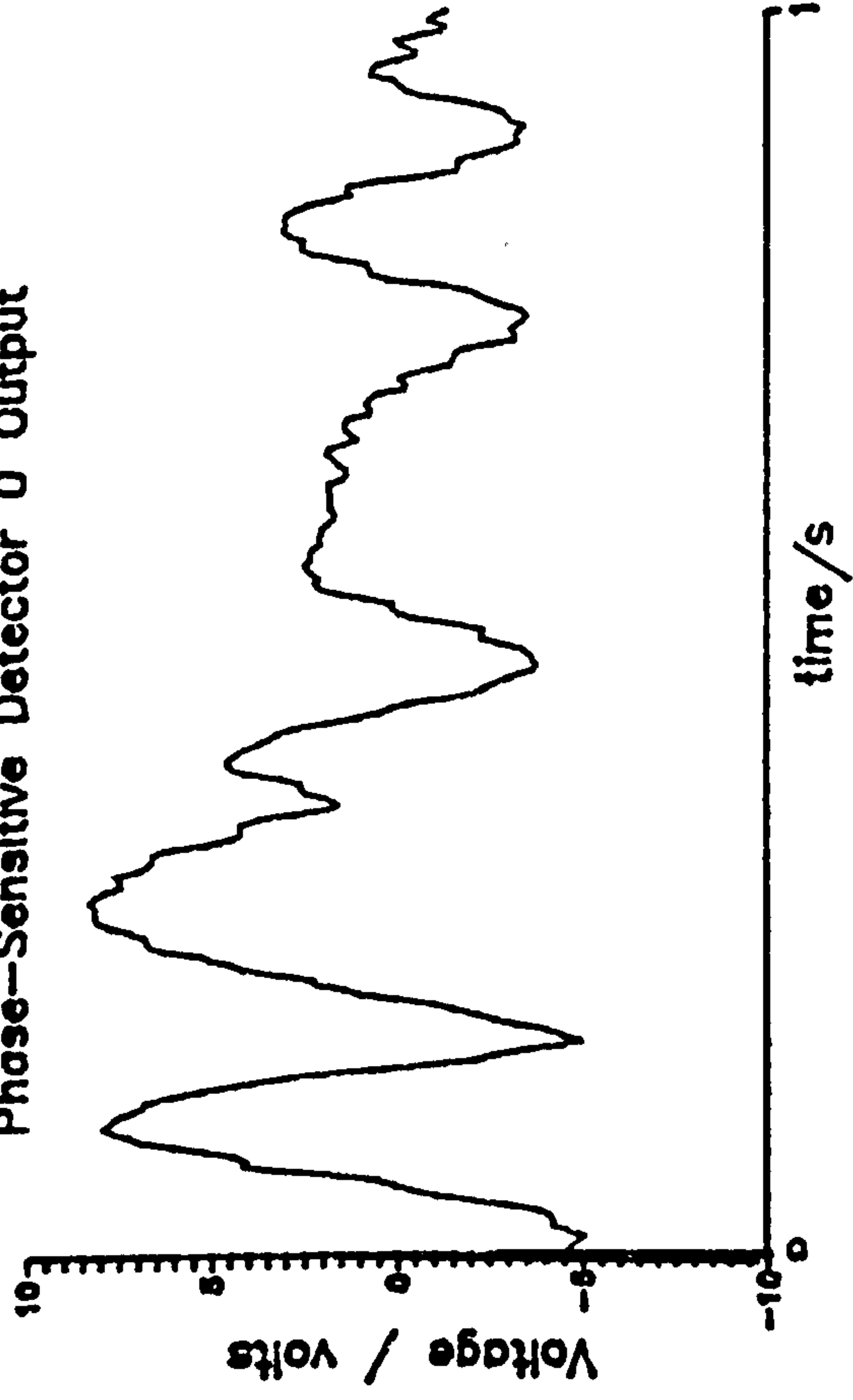
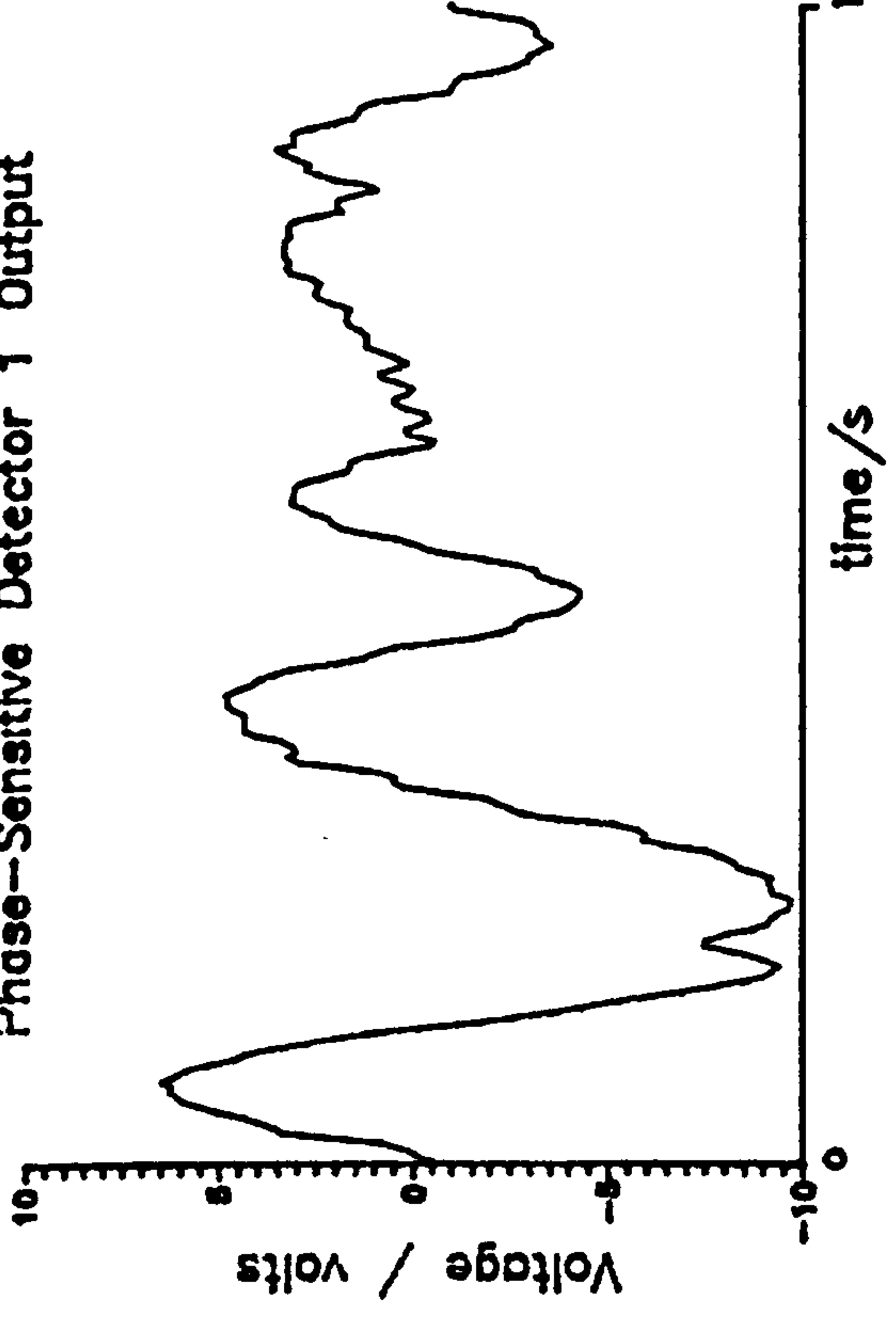


Figure 4.24(a): PSD outputs and calculated amplitude and phase for zeolite particles dispersed in ethylene glycol with no electric field applied.

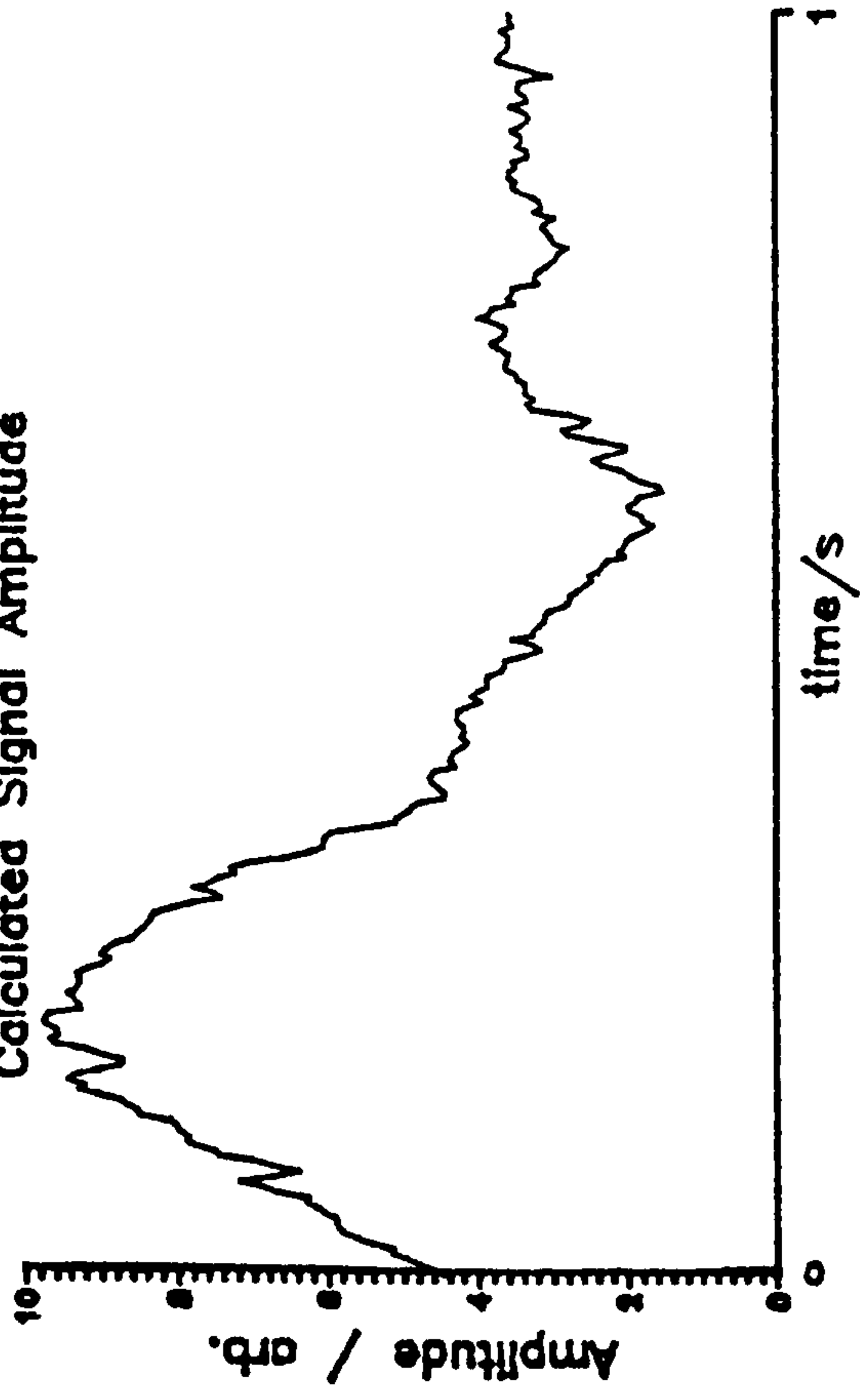
Phase--Sensitive Detector 0 Output



Phase--Sensitive Detector 1 Output



Calculated Signal Amplitude



Calculated Signal Phase

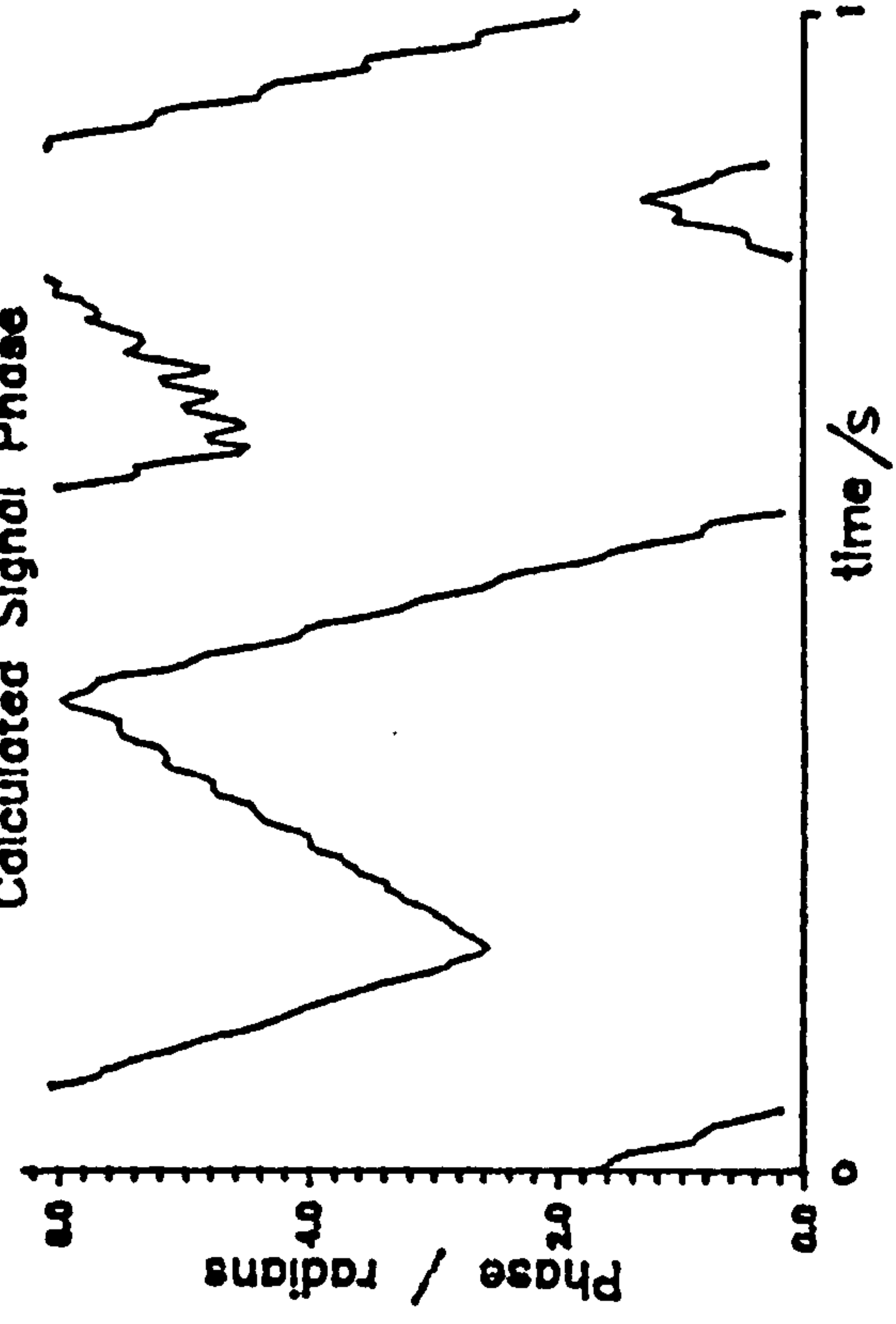
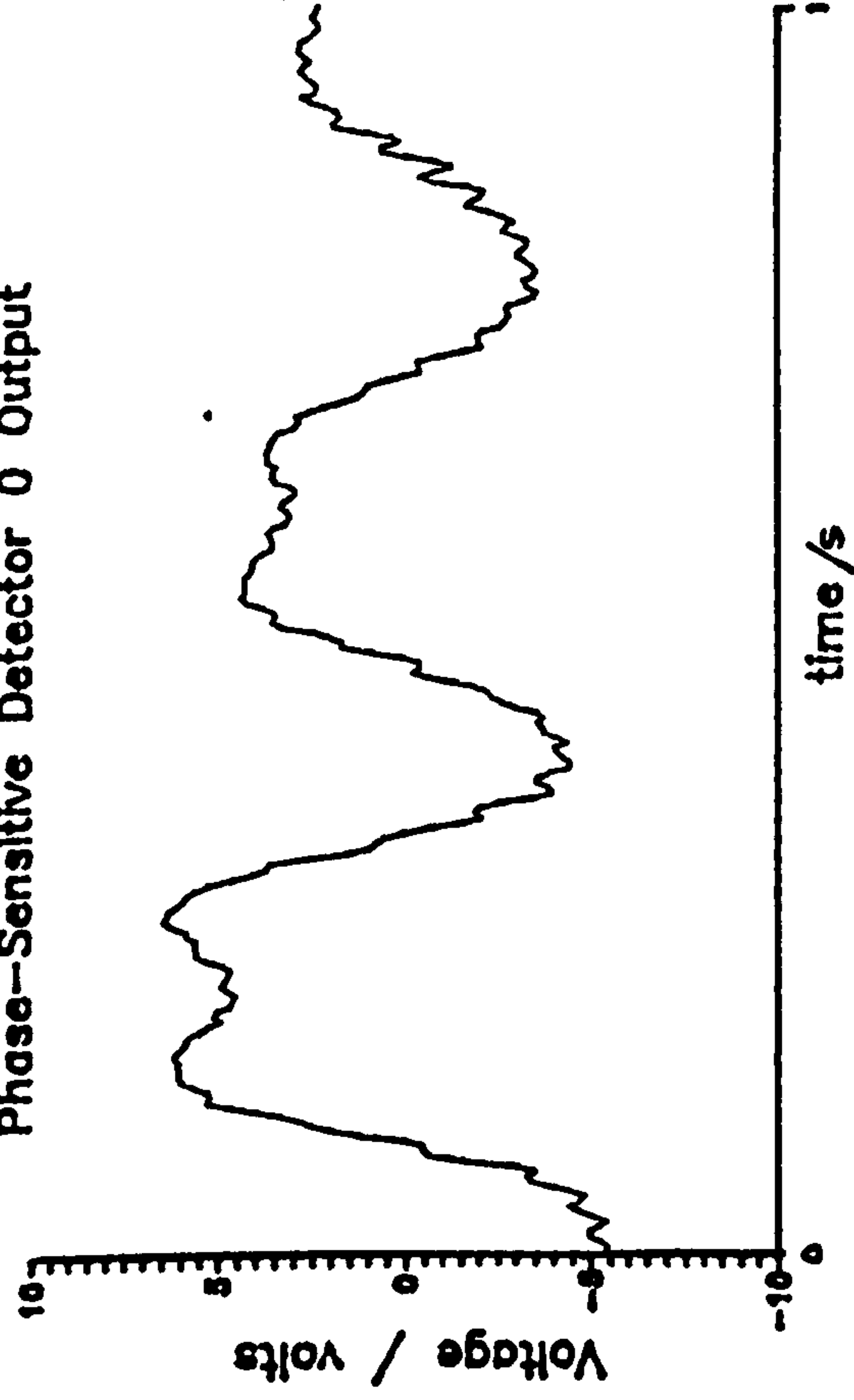


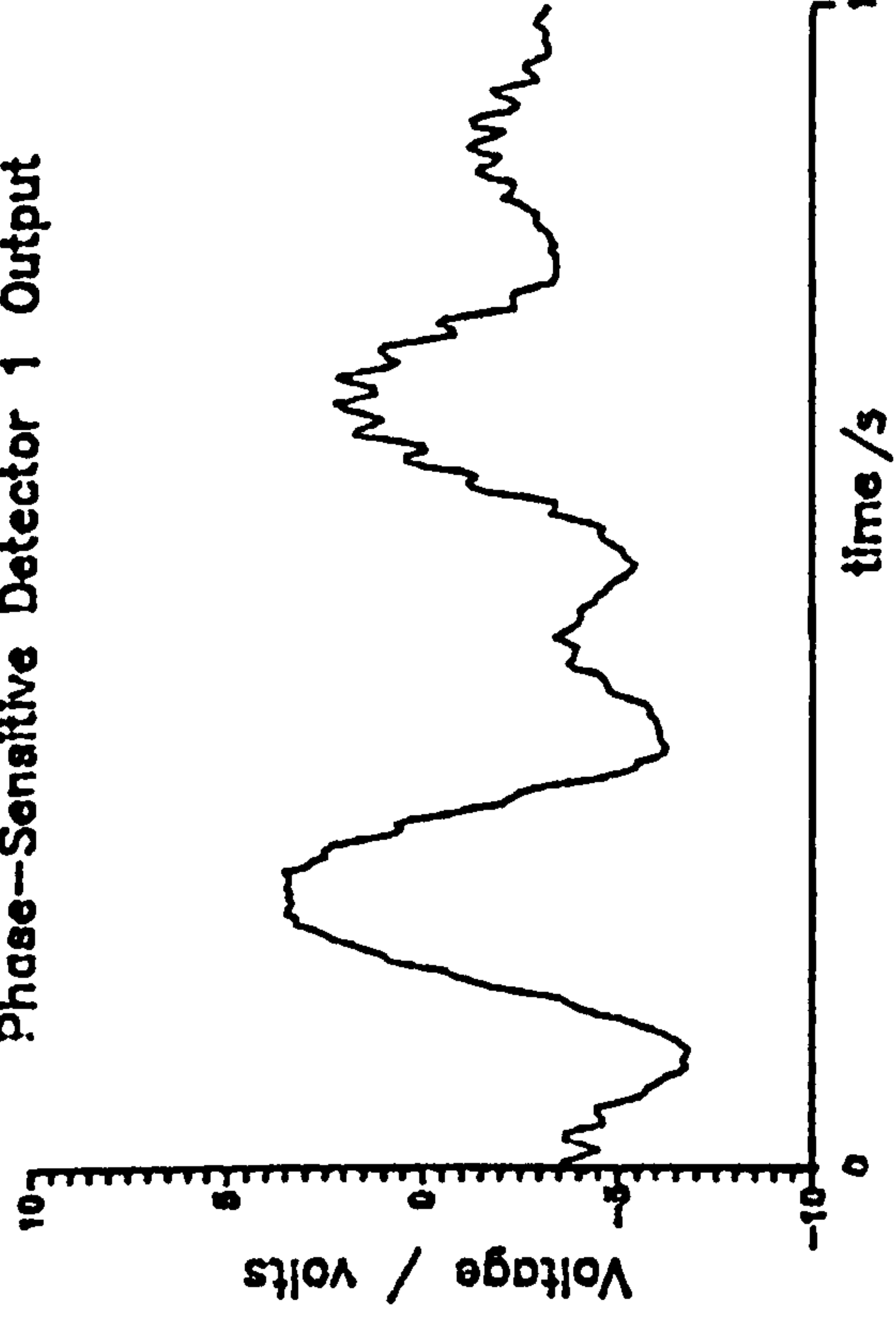
Figure 4.24(b): PSD outputs and calculated amplitude and phase for zeolite particles dispersed in ethylene glycol subjected to a square wave electric field.



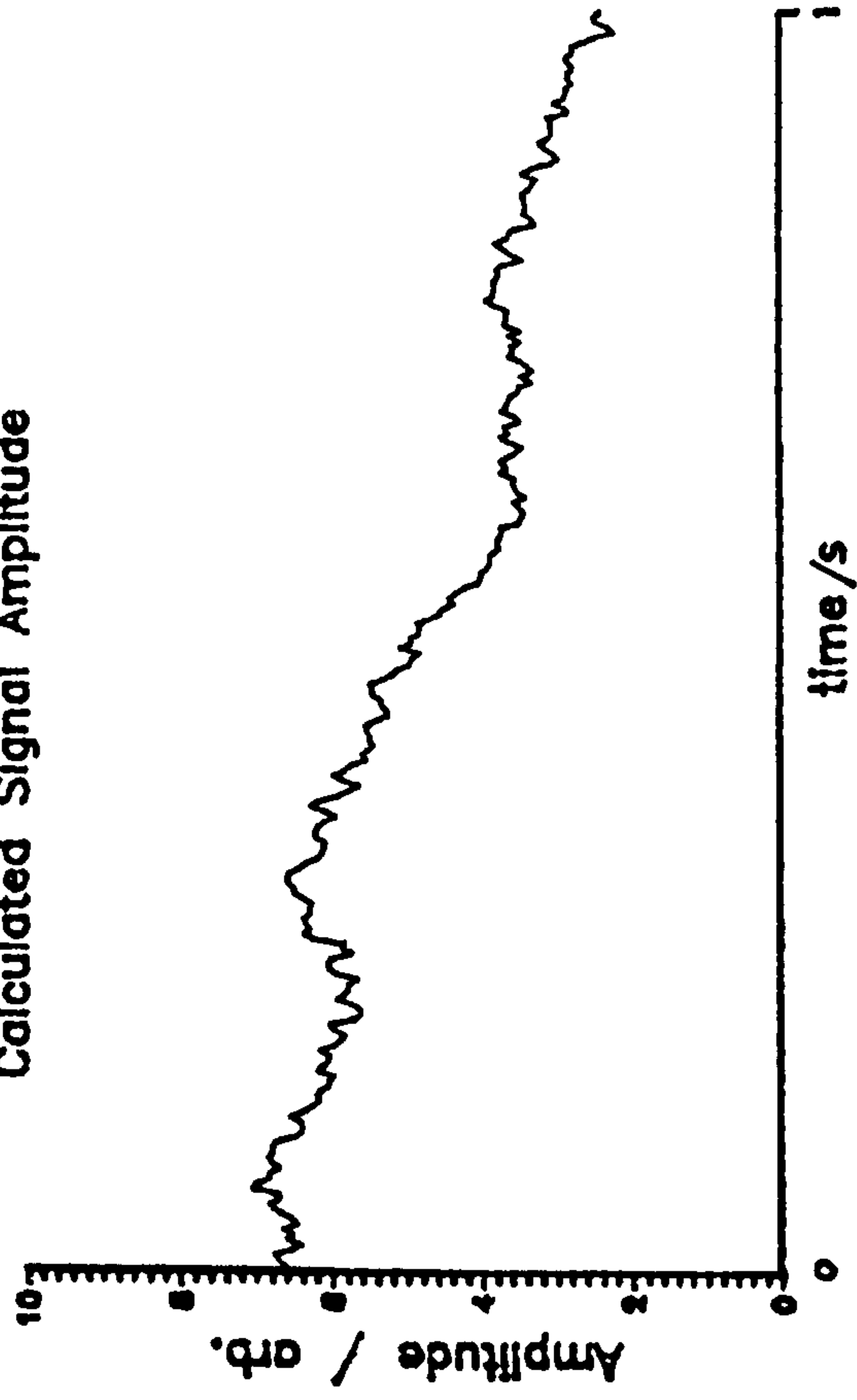
Phase-Sensitive Detector 0 Output



Phase-Sensitive Detector 1 Output



Calculated Signal Amplitude



Calculated Signal Phase

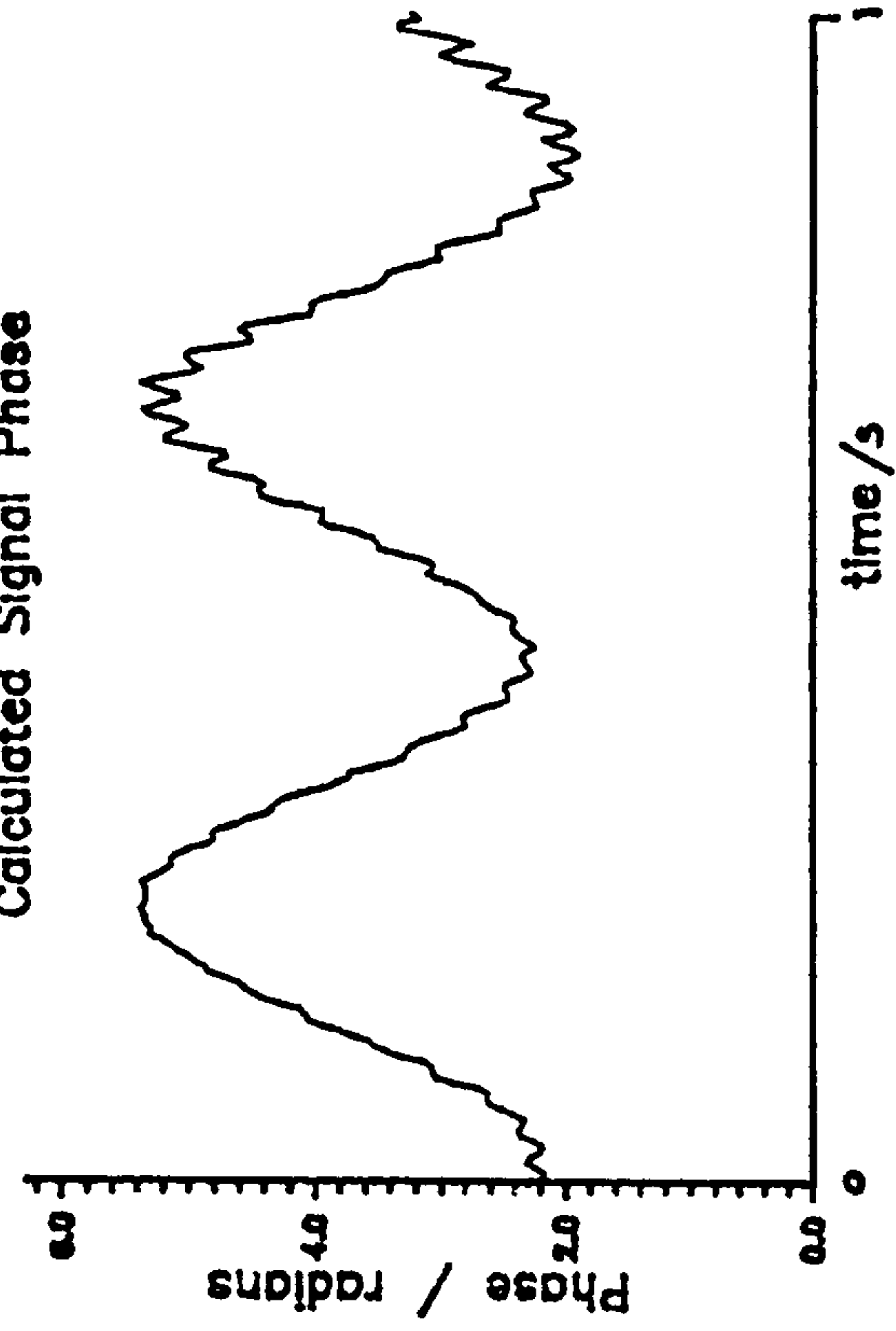


Figure 4.24(c): PSD outputs and calculated amplitude and phase for zeolite particles dispersed in ethylene glycol subjected to a sinusoidal field.

can be considered instantaneous. It can also be seen that as the amplitude of the scattered light signal diminishes, the noise contribution in the phase information increases. Case (c) provides a good example of a sinusoidal phase change occurring for particles undergoing electrophoresis in the presence of an applied sinusoidal field. In this case, the amplitude of the oscillation of the particles is approximately half a fringe spacing equivalent. Thus, qualitatively at least, the PALS hardware so far described yields the expected time domain information. Figures 4.25 (a) and 4.25 (b) show the time domain signals for an aqueous polystyrene latex dispersion, both in the absence of an electric field, (a), and in the presence of a sinusoidal field, (b). The diffusive contribution is more prominent in this case than for the zeolite dispersion, since the viscosity of ethylene glycol is ~ 20 times that of water.

Figure 4.26 shows the experimental AWPDP function obtained for a dilute dispersion of polystyrene latex particles in dilute aqueous potassium chloride solution. The least squares fit is superimposed on the experimental data. As can be seen, excellent agreement is found between theory and experiment. However, the magnitude of the determined mobility,  $1.08 \times 10^{-8} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$ , is lower than determined using the classical frequency analysis processing scheme. Figure 4.27 shows the frequency spectrum obtained for the same dispersion. This spectrum yields a mobility of  $3.66 \times 10^{-8} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$ . It can be

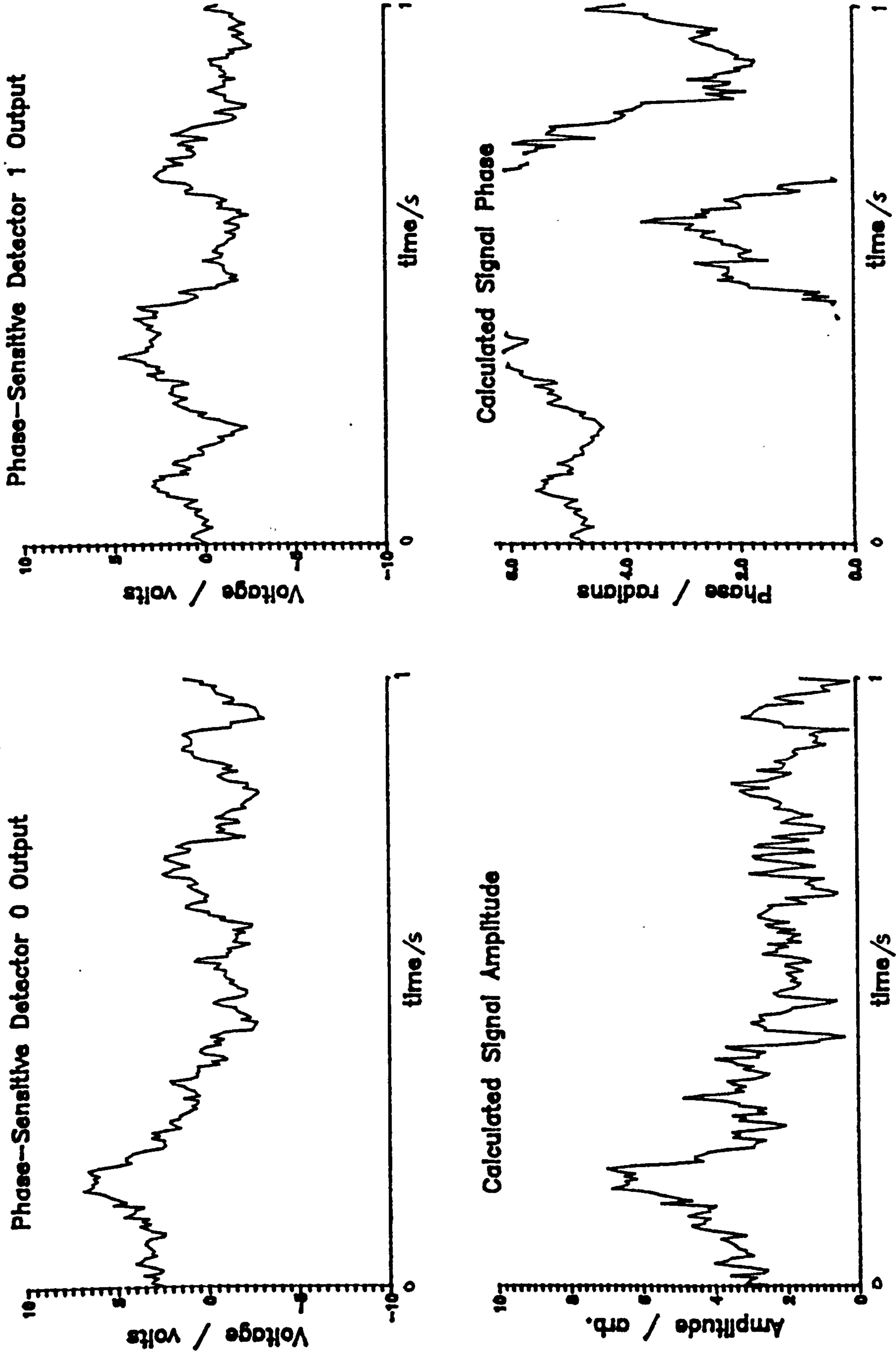
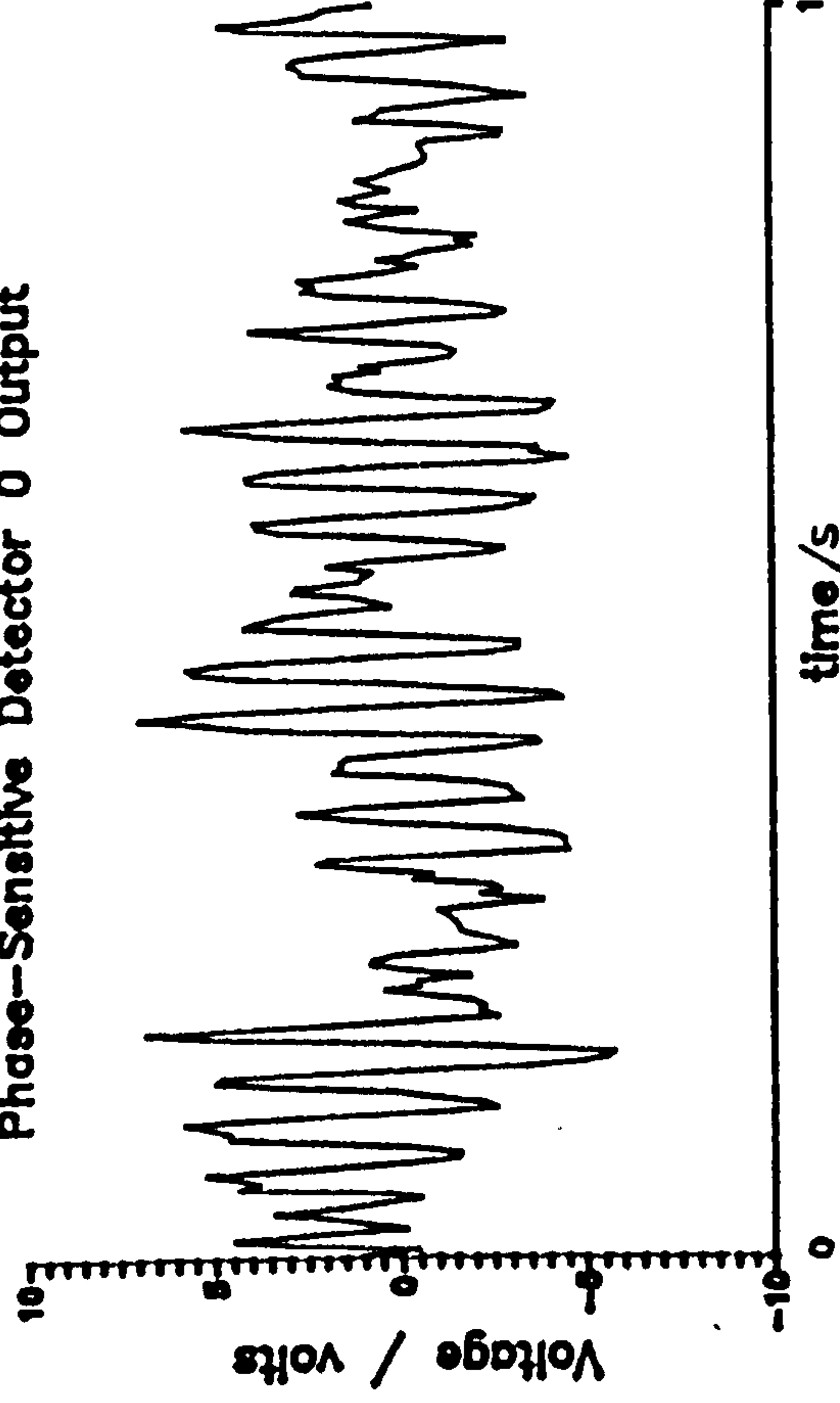
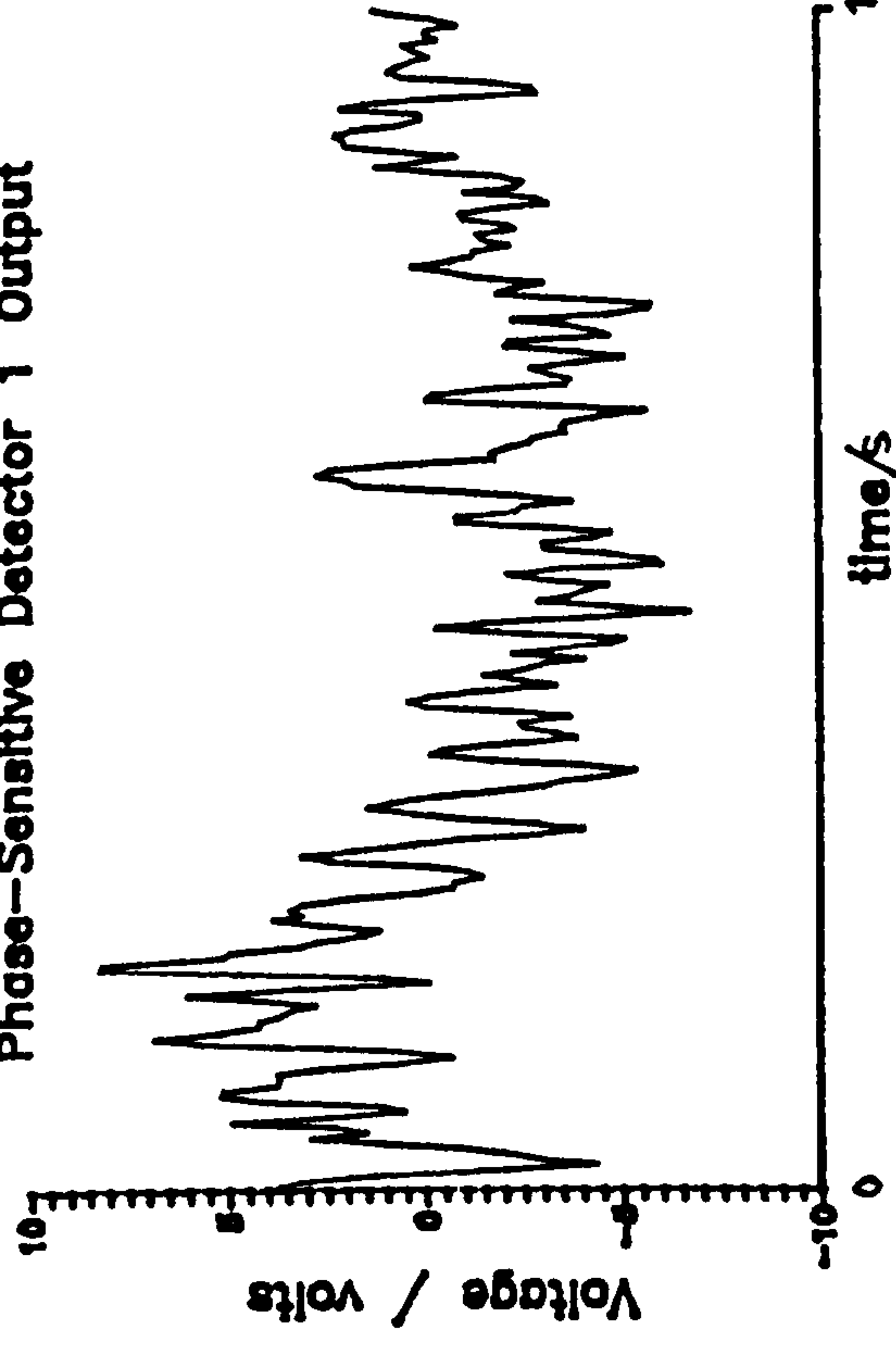


Figure 4.25(a): PSD outputs and calculated amplitude and phase for polystyrene latex particles dispersed in water with no electric field applied.

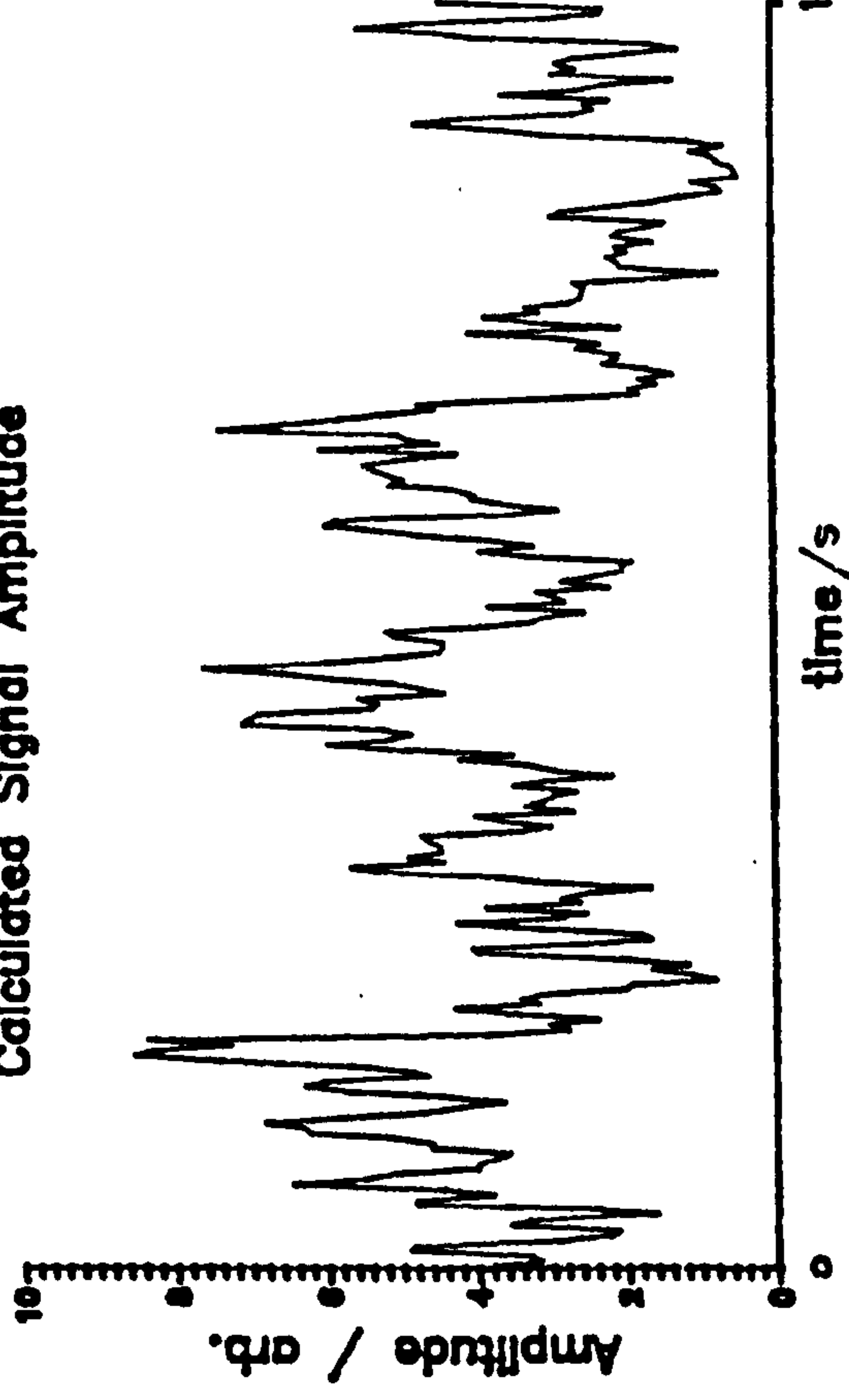
Phase--Sensitive Detector 0 Output



Phase--Sensitive Detector 1 Output



Calculated Signal Amplitude



Calculated Signal Phase

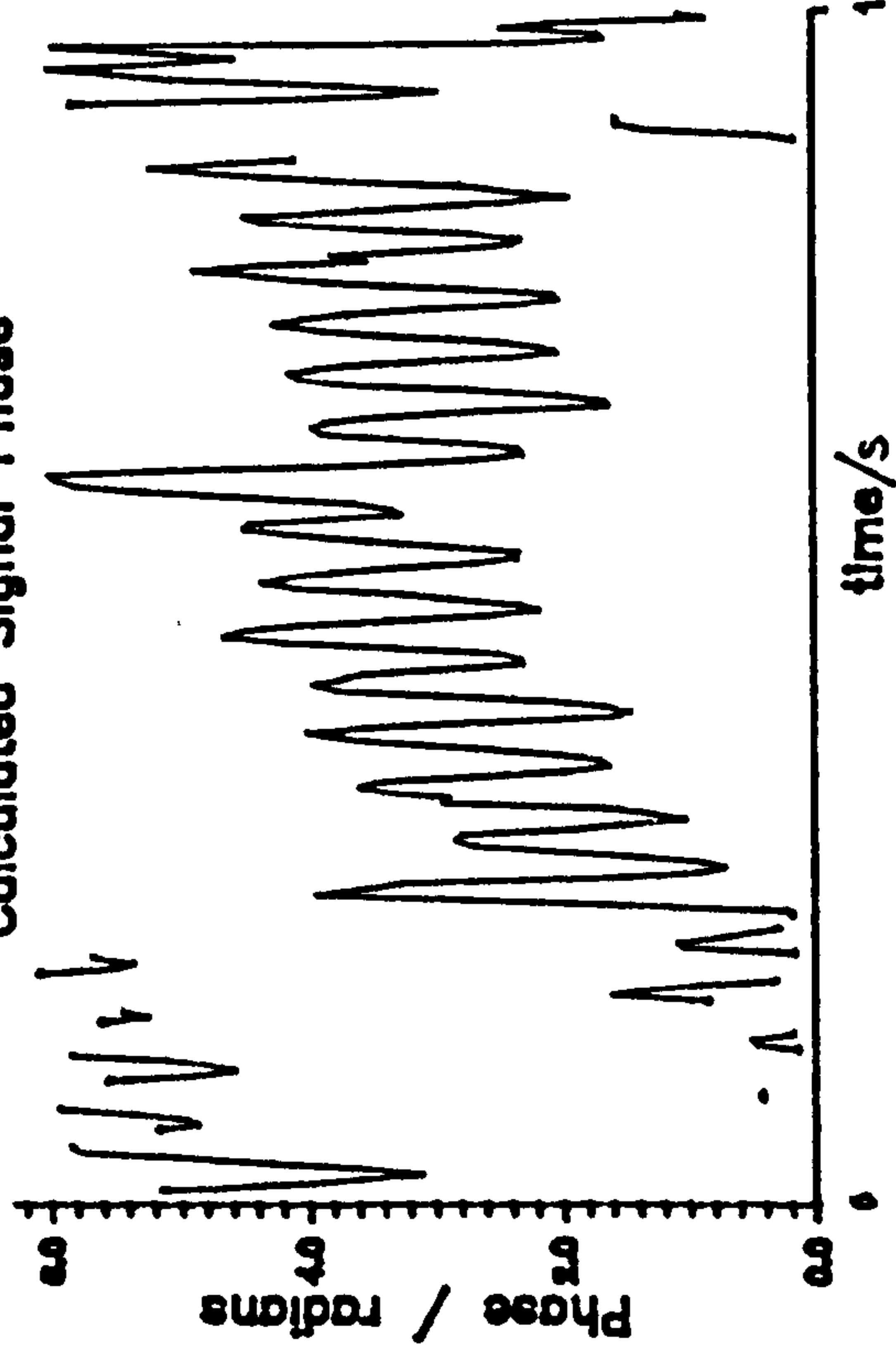


Figure 4.25(b): PSD outputs and calculated amplitude and phase for polystyrene latex particles dispersed in water subjected to a sinusoidal electric field.

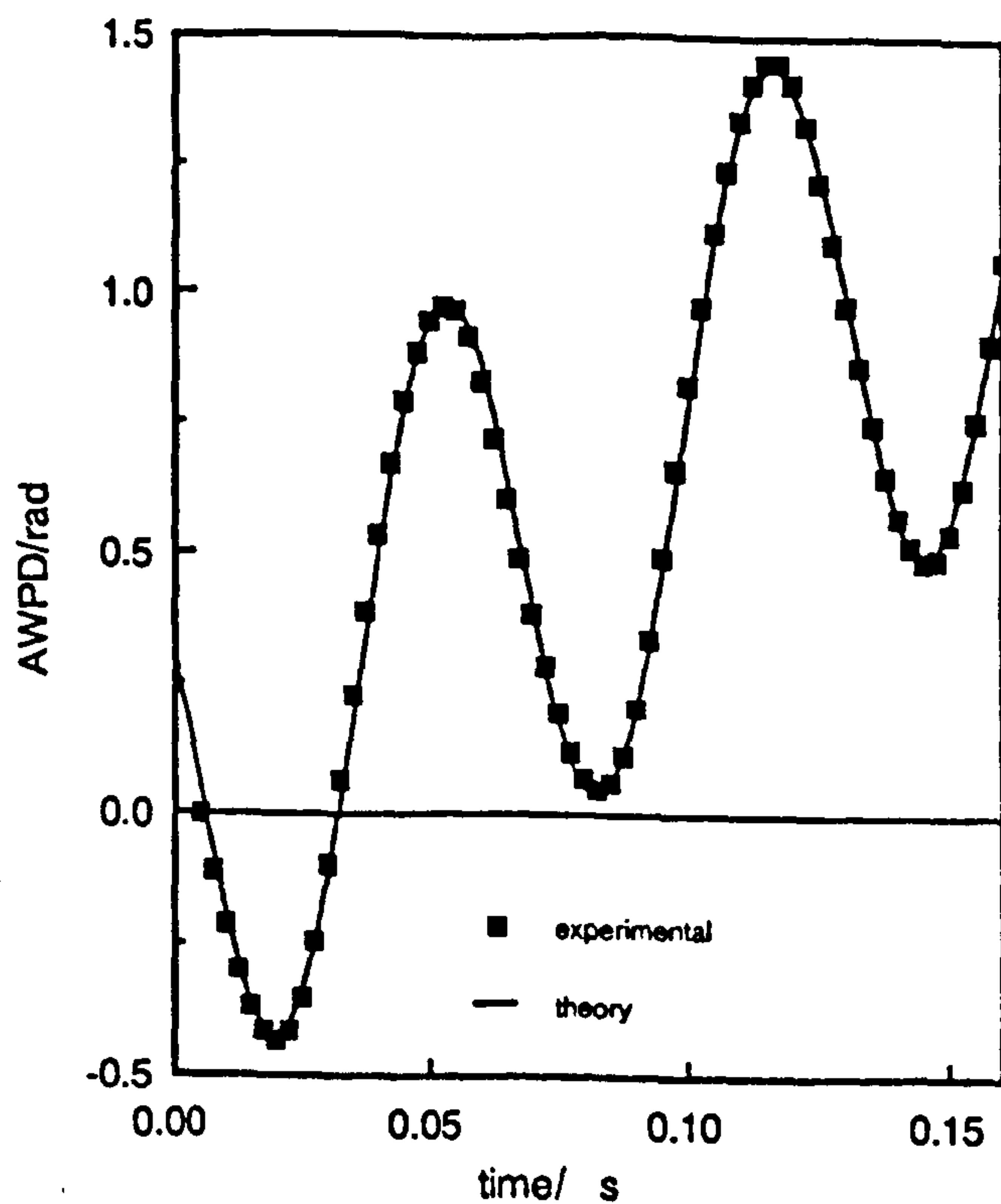


Figure 4.26: AWPDP function obtained for a dilute dispersion of polystyrene latex particles in dilute KCl (aq) subjected to a sinusoidal electric field.

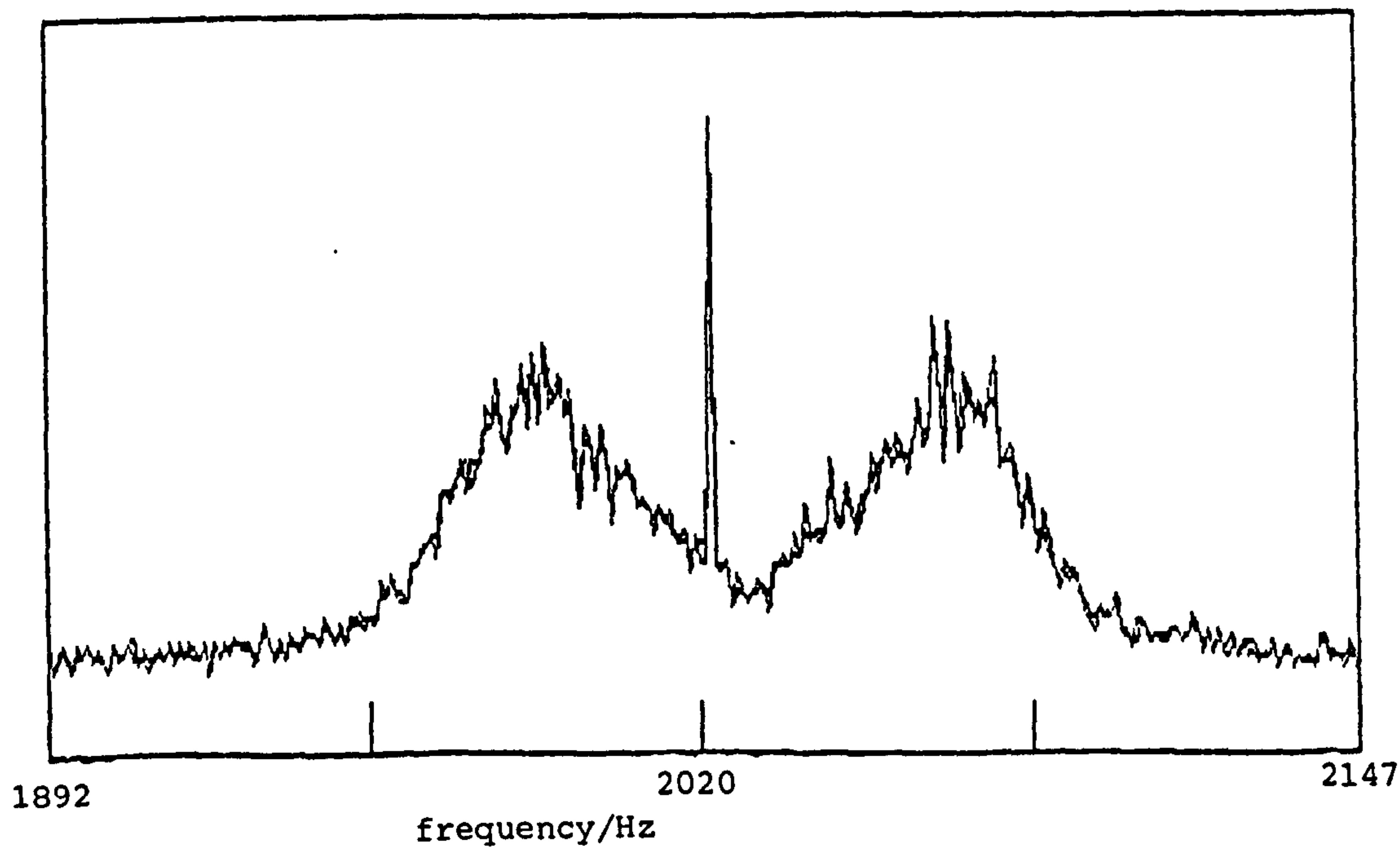


Figure 4.27: Moving fringe LDE spectrum for a dilute dispersion of polystyrene latex particles in dilute KCl (aq) subjected to a square wave electric field.



seen that there exists a very sharp frequency component at  $\sim 2020$  Hz, which arises through the reasons already described in § 4.2.2. Initially, it was thought that the underestimation of mobilities from the AWPD function, compared with spectral analysis, arose as a result of the centre frequency interference. However, the interference can be minimised, as described earlier, until no detectable interference is present. The following results from a study of  $\sim 200$  nm diameter rutile ( $\text{TiO}_2$ ) particles dispersed in  $10^{-3}$  mol dm $^{-3}$  aqueous potassium chloride solution help to highlight this problem. The mobility of the rutile particles was determined using classical LDE and from the AWPD function, as a function of the attenuation of the LF input signal to the SSBM. The potentiometer is scaled from 2 (maximum attenuation) to 14 (no attenuation). Figure 4.28 shows that the PALS-determined mobility,  $\mu_{\text{PALS}}$ , is consistently lower than the LDE-determined mobility,  $\mu_{\text{LDE}}$ . At the optimal attenuation setting (3 to 4), no centre frequency interference was detectable, hence another reason is required to explain the underestimation of velocity information using the phase analysis scheme.

Figure 4.29 shows how  $\mu_{\text{PALS}}$  for the rutile dispersion varies as a function of applied field strength and frequency. It is apparent that there exists a "mobility window" where  $\mu_{\text{PALS}}$  and  $\mu_{\text{LDE}}$  agree, within experimental uncertainty. However, the use of too low or too high field strengths leads to underestimation of  $\mu_{\text{PALS}}$ . The



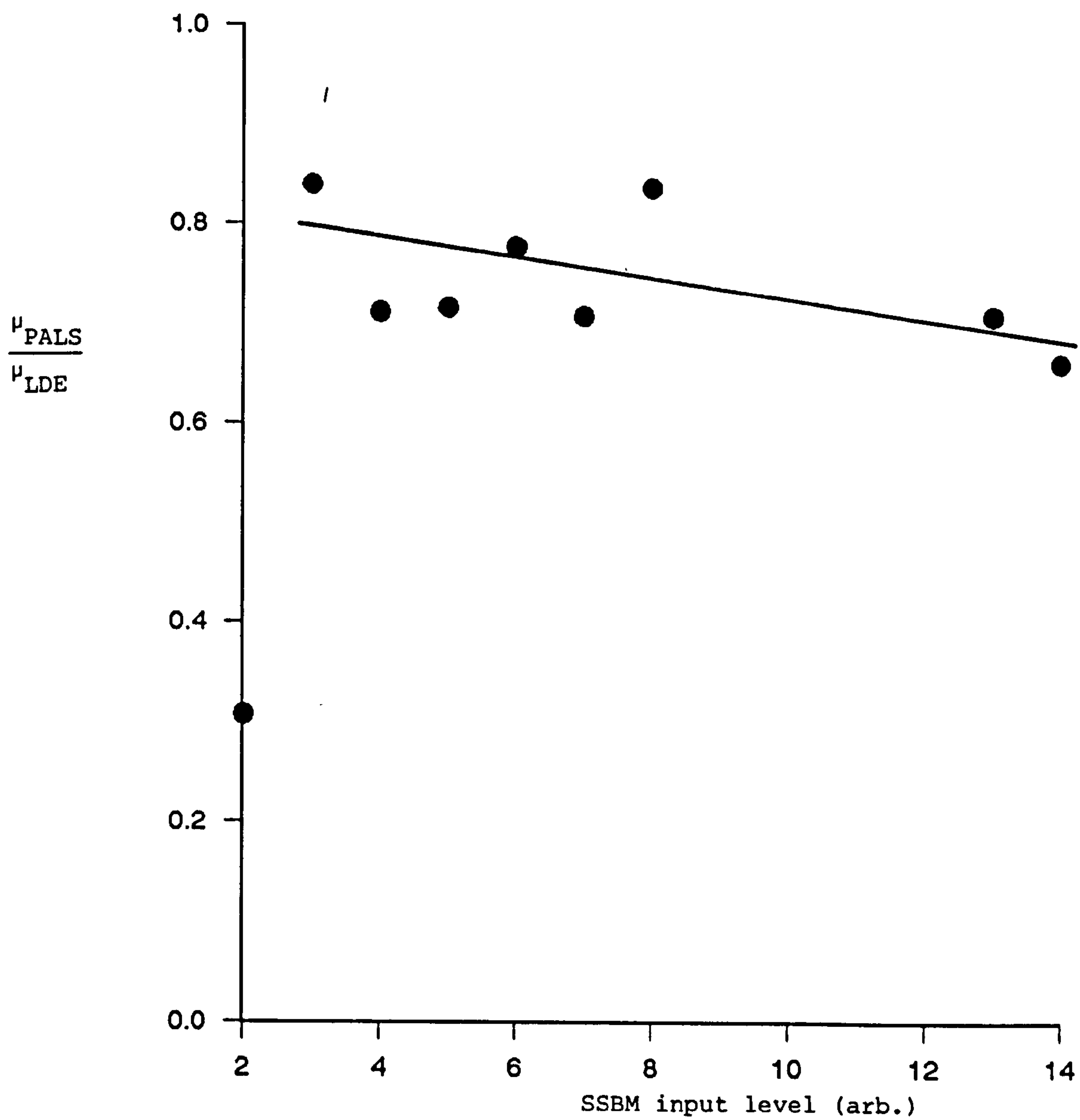
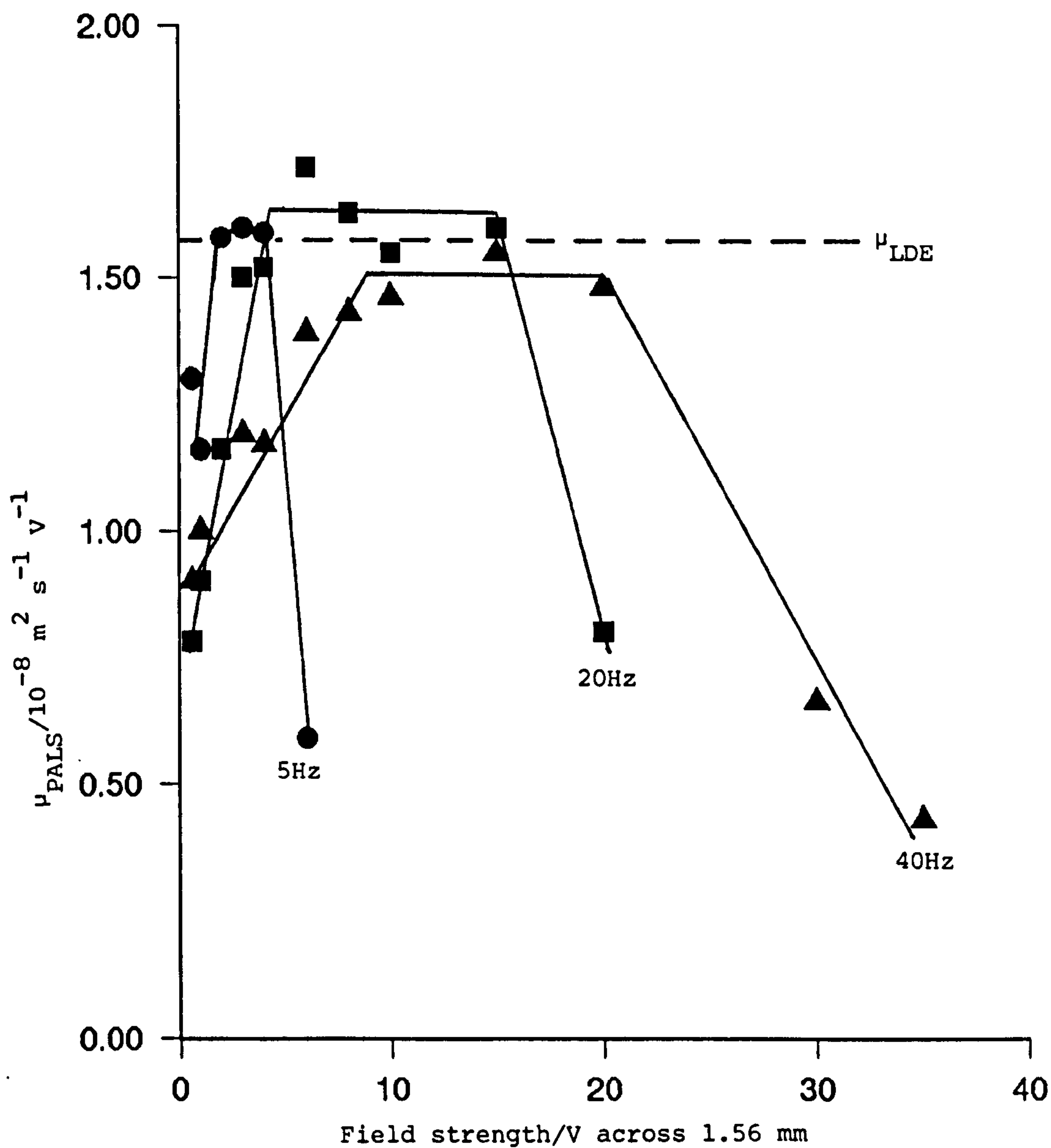


Figure 4.28: Ratio of  $\mu_{\text{PALS}}$  to  $\mu_{\text{LDE}}$  for rutile dispersed in  $10^{-3} \text{ mol dm}^{-3}$  KCl (aq) as a function of SSBM input level.



**Figure 4.29:** Mobility of rutile dispersed in  $10^{-3} \text{ mol dm}^{-3}$  KCl (aq) determined by PALS as a function of applied field strength at various field frequencies.

problem of too high field strengths arises through the limited bandwidth of the p.s.ds, and the problem of too low field strengths probably arises through the p.s.d. circuits being rather noisy. The p.s.ds are mid-1970's technology and, hence, their signal-to-noise ratio performance characteristics for weak, slowly changing signals are somewhat poor. Clearly, the underestimation problem at low field strengths is undesirable, especially for the study of non-polar dispersions, where it is hoped that relatively low field strengths could be employed. The solution to this problem is described later.

#### 4.2.10 PRELIMINARY STUDIES OF NON-POLAR DISPERSIONS USING PALS

The apparatus so far described is capable of applying sinusoidal signals up to 10V across the electrodes. It was realised that this would not be enough for studying non-polar dispersions with small mobilities ( $10^{-10}$  to  $10^{-11} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$ ). Hence, an amplifier was built (by the University of Bristol Chemistry Department Electronics Workshop) to amplify the electrode signal by a factor of  $\sim 10$ . Figure 4.30 shows the output voltage of the amplifier as a function of the input voltage at a fixed frequency of 50 Hz. Figure 4.31 shows the frequency response of the amplifier at a fixed input voltage (0.0971 V rms). Thus, it is readily seen that the amplifier has a uniform gain of 11.11 (+ 20.9 dB) for input signals

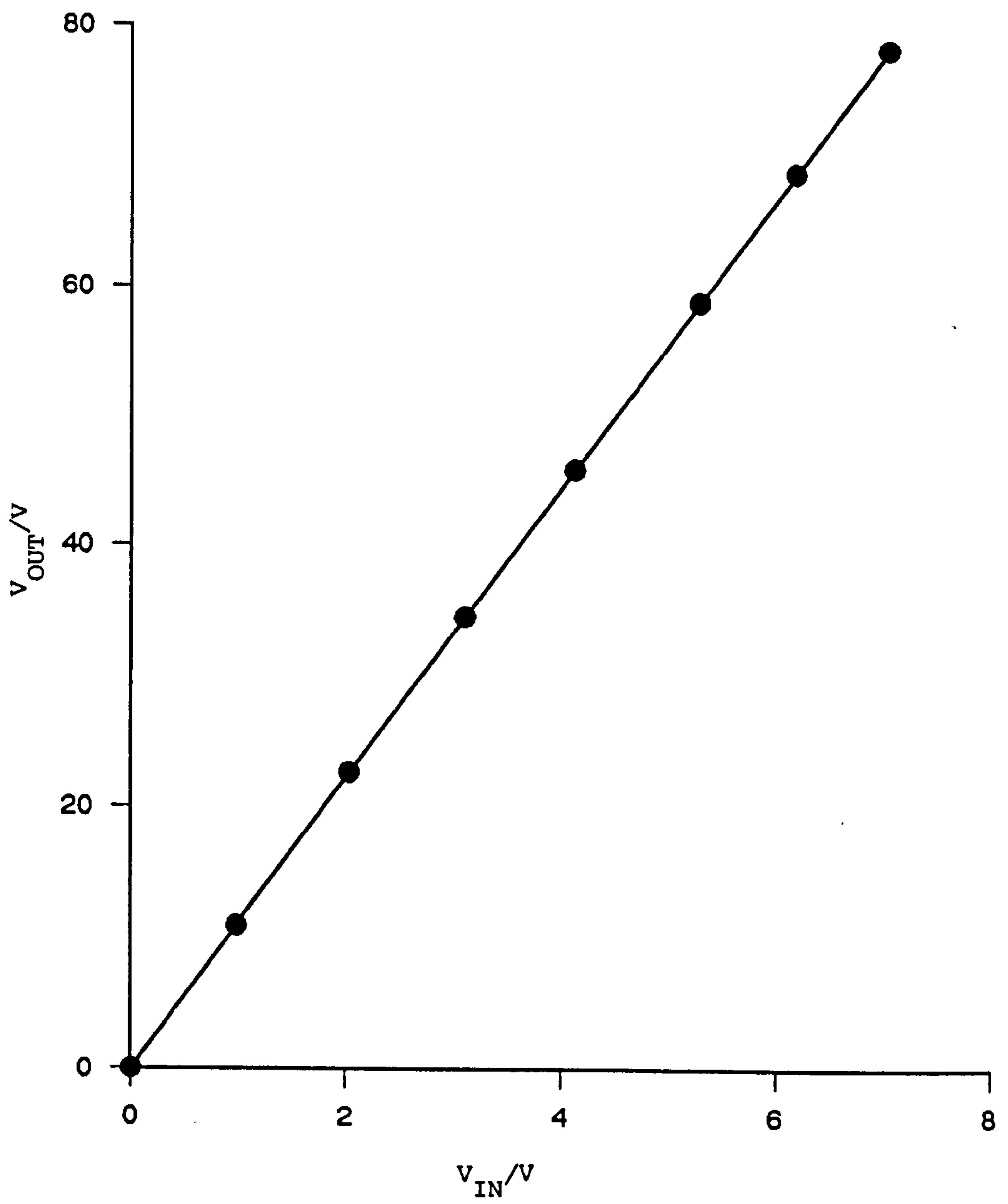


Figure 4.30:  $V_{OUT}$  VS.  $V_{IN}$  for x10 amplifier at 50 Hz.

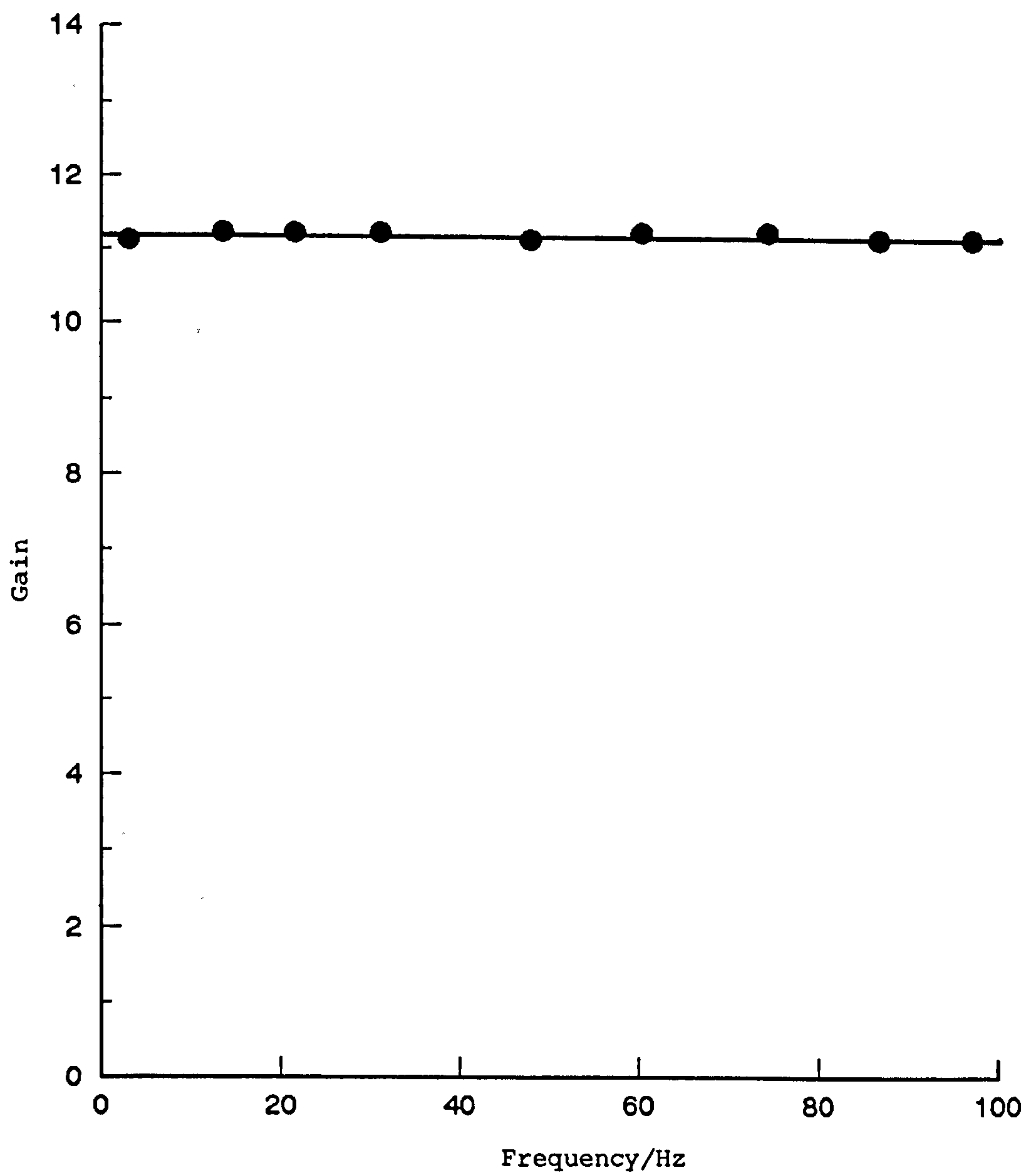


Figure 4.31: Graph to illustrate flat frequency response of x10 amplifier for signals up to 100 Hz.

up to 10V in amplitude and 100 Hz in frequency. The amplifier was found to work equally well for square waves and sinusoidal waves. The input signal is obtained from the TG-102 generator, and the output is connected to the electrodes, enabling maximum voltages of  $\pm 111\text{V}$  to be applied to dispersions.

A dispersion of  $\sim 200$  nm diameter rutile particles in p-xylene (1, 4-dimethylbenzene) was chosen as the system to test the ability of PALS to determine mobilities of non-polar dispersions. Figure 4.32 shows the AWPDP function (experimental data and least squares fit) for a dilute rutile/p-xylene system. Thus, qualitatively at least, good quality AWPDP functions can be obtained for non-polar dispersions. A quantitative study was performed by measuring  $\mu_{\text{PALS}}$  and  $\mu_{\text{LDE}}$  for the rutile/p-xylene system both as a function of trace water content at fixed particle concentration and as a function of particle concentration at fixed water content. In all cases, the water concentration was determined using the method described in the next chapter, after mobility measurements had been made. The p-xylene was dried in the manner described in the next chapter. Figure 4.33 shows how  $\mu_{\text{PALS}}$  and  $\mu_{\text{LDE}}$  vary as a function of trace water concentration (for PALS mobilities, applied field = 32 V across 1.56 mm at 46 Hz; for LDE mobilities applied field = 32 V across 1.56 mm at 4.6 Hz). Qualitatively,  $\mu_{\text{PALS}}$  shows the same trend as  $\mu_{\text{LDE}}$ , but the former is underestimated by  $\sim 50\%$ . In figure 4.34,  $\mu_{\text{PALS}}$  is also underestimated.



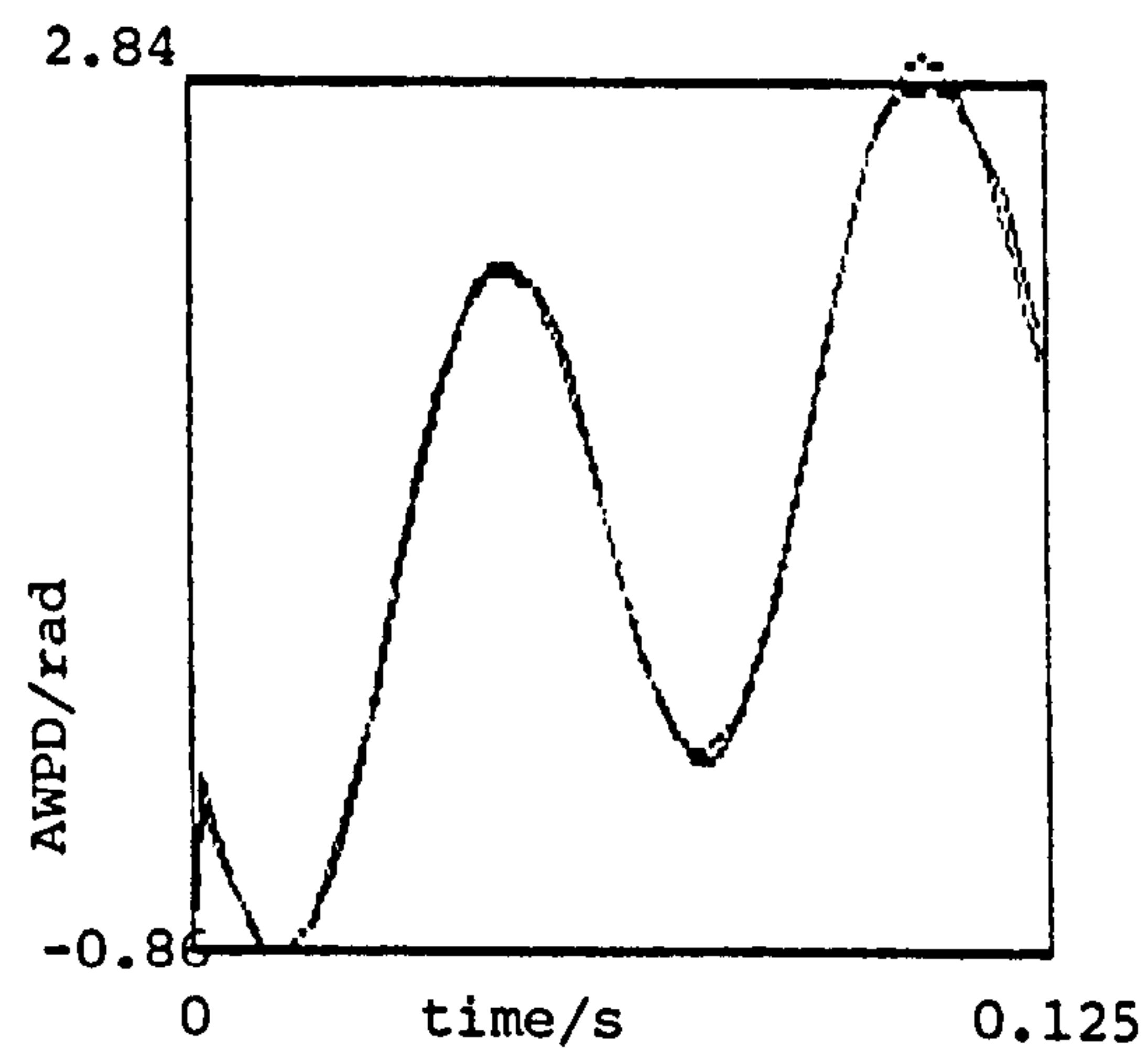


Figure 4.32: AWPD for rutile particles dispersed in p-xylene.  
Field strength = 50 V across 1.56 mm at 16 Hz.

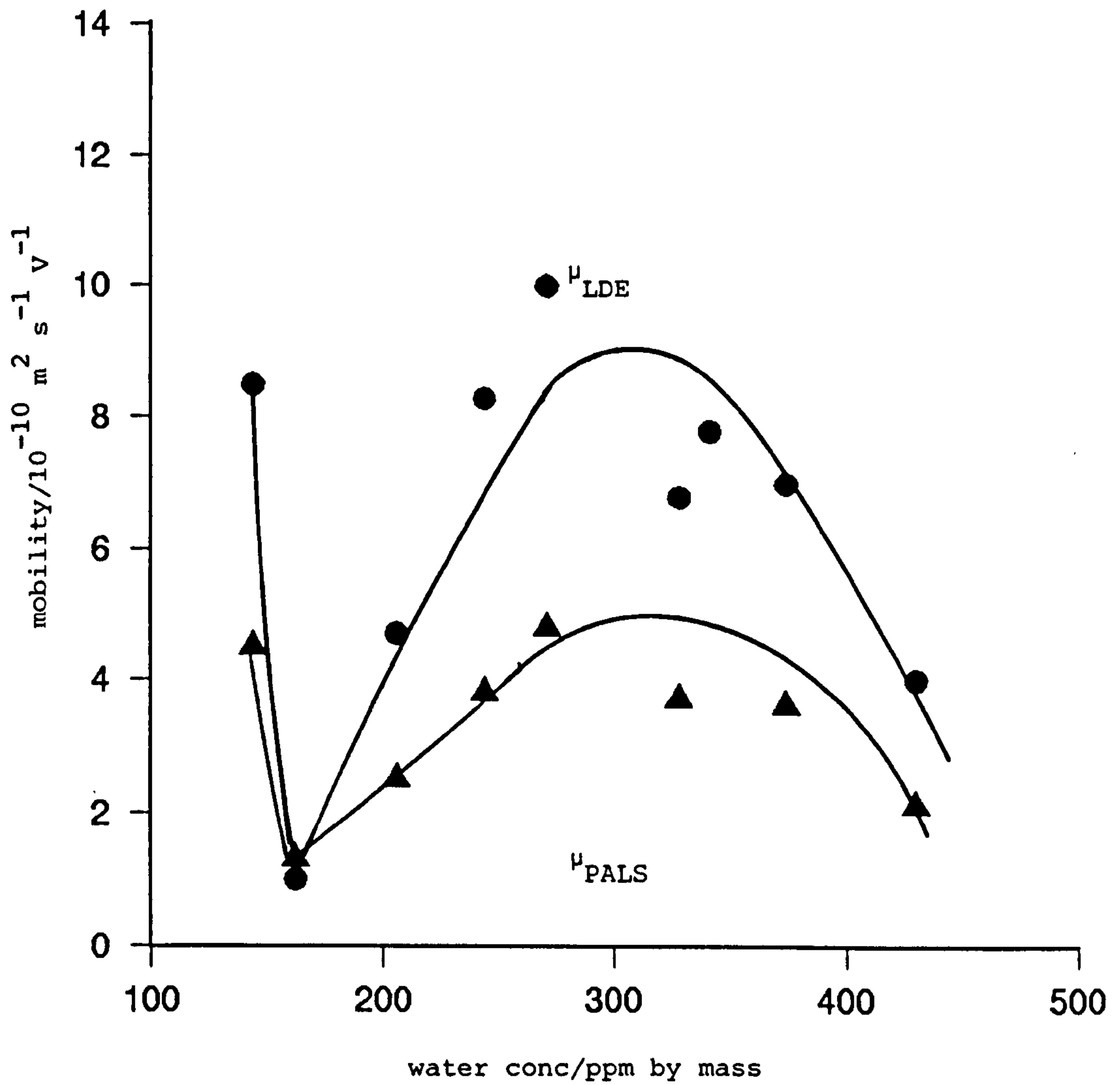


Figure 4.33: Comparison between  $\mu_{\text{PALS}}$  and  $\mu_{\text{LDE}}$  for rutile dispersed in p-xylene as a function of concentration of water in bulk phase.

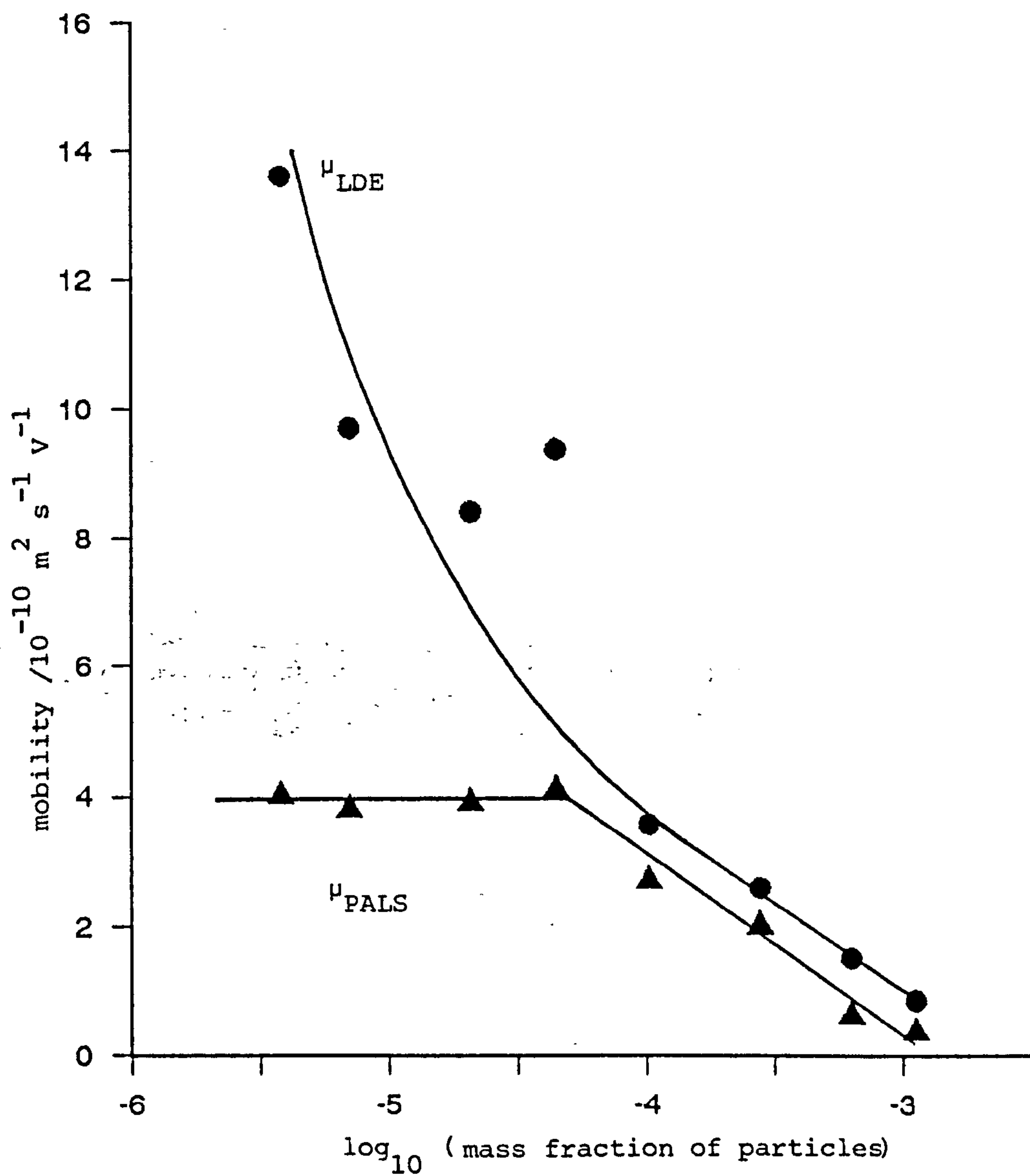


Figure 4.34: Comparison between  $\mu_{PALS}$  and  $\mu_{LDE}$  for rutile dispersed in p-xylene as a function of mass fraction of particles.

The underestimation becomes particularly severe as the mass fraction of rutile in the dispersion becomes very small ( $10^{-5}$  to  $10^{-6}$  mass fraction). It is believed that the underestimation therefore arises through the rather poor signal-to-noise ratio of the p.s.d. units used. If this were the case, then modern, state-of-the-art p.s.d. units would overcome this problem.

#### 4.2.11 INCORPORATION OF A LOCK-IN AMPLIFIER FOR PHASE-SENSITIVE DETECTION

A lock-in amplifier (LIA) is essentially a phase-sensitive detector, combined with pre-detector amplification and filtering circuitry, and post-detector filtering circuitry. A dual phase LIA is an apparatus containing two p.s.ds, driven in quadrature. Thus, a dual phase LIA is, in principle, suited for use in a PALS apparatus. A state-of-the-art dual phase LIA (Princeton Applied Research Corporation Model 5210) was obtained in an attempt to remove the noise problem described in the previous section. This LIA was chosen primarily for its excellent signal-to-noise characteristics (5 nV per root Hz at 1 kHz) and its sensitivity (100 nV to 3V input range yielding 10V full scale output). The 5210 LIA also provides pre- and post-detector filtering. The pre-detector filtering may operate in band pass, low pass, notch or flat modes. Line filters are provided which can filter out frequencies at 50 Hz and 100 Hz. The use of band

pass filtering, line filtering and a floating ground for the input signal return allow the scattered light signal to be suitably conditioned for PALS processing. The pedestal signal and any mainsborne interference are removed, and only the frequency information centred about the reference frequency is amplified. Hence, the 5210 LIA is the only piece of hardware required between the PM tube anode and the ADC inputs. An internal programmable oscillator serves to provide the reference frequency. This signal is available at the front of the LIA via a BNC connector, thereby providing the LF input signal for the SSBM. The frequency of the oscillator can be accurately determined using the LIA's integral frequency counter. An external signal may be used as the reference instead. The frequency counter determines the frequency of the signal currently being used as the reference. The internal oscillator produces the same pre-programmed frequency, irrespective of the source of the reference.

The post-detector filtering provides various levels of smoothing of the p.s.d. output signals, the minimum filtering time constant available being 1 ms. This results in the two p.s.d. outputs having bandwidths of  $\sim 250$  Hz. For the generation of AWPD and AWPS functions, this is not a problem. However, if the correlation function for either of the p.s.d. outputs (i.e. the real or imaginary components of the scattered light signal relative to the reference signal) is required, the limited bandwidth will tend to flatten the



function for correlation times of the order of a few milliseconds. The 5210 LIA provides an unfiltered p.s.d. output for one of the channels, more suited to the construction of correlation functions. However, this signal has a maximum output of 1V, rather than 10V. Thus, before this signal reaches the ADC, it is amplified by a factor of 10 using one of the EF5-01 filter units described earlier.

The 5210 LIA is fully programmable by computer, using either an IEEE or RS-232 communications interface. For the PALS apparatus the RS-232 interface is employed. Further details can be found in Appendix 2 and in the 5210 LIA Manual<sup>116</sup>.

The amplified and filtered scattered light signal is available at the front panel of the 5210 LIA for monitoring purposes. This signal can be fed into the 3582A spectrum analyser. The availability of a frequency spectrum of the scattered light signal during a PALS experiment is very useful for detecting excessive thermal convection or sedimentation/creaming of the sample, and for assessing the general quality of the sample with regard to suitability for analysis by PALS. Furthermore, by switching the electrode drive signal generator from sinusoidal output to squarewave output, the PALS apparatus can operate in the conventional frequency analysis LDE mode.



#### 4.2.12 ELECTRODE DRIVE SIGNAL CHARACTERISATION

For the earlier version of the apparatus, the electrode drive amplitude was determined using an oscilloscope and the electrode drive frequency was determined using the spectrum analyser. However, the incorporation of the 5210 LIA allows the electrode drive frequency to be measured using the 5210's frequency counter. For electrode drive amplitudes below 10 V, the amplitude can be determined using one of the A/D channels of the DT2818 ADC board. Hence, the PALS software can measure the electrode drive amplitude itself. However, amplitudes above 15 V may damage the ADC and amplitudes above 10 V will appear to be 10 V. Thus, an electronic conditioning unit was devised which attenuates the signal presented to the ADC by the factor of 10 (i.e. 20 dB attenuation) if the electrode drive signal amplitude exceeds 10 V. A TTL-compatible signal from this unit instructs the PALS software whether or not the signal is being attenuated. If the amplitude is below 10 V, then the signal is unaffected. Hence, the apparatus can automatically measure amplitudes ranging from 0 to 100 V without damage occurring to the ADC. LED indicators on the front of the conditioning unit inform the user of the state of the TTL signal. Figure 4.35 shows schematically how the unit is incorporated into the apparatus. The conditioning unit was calibrated in the

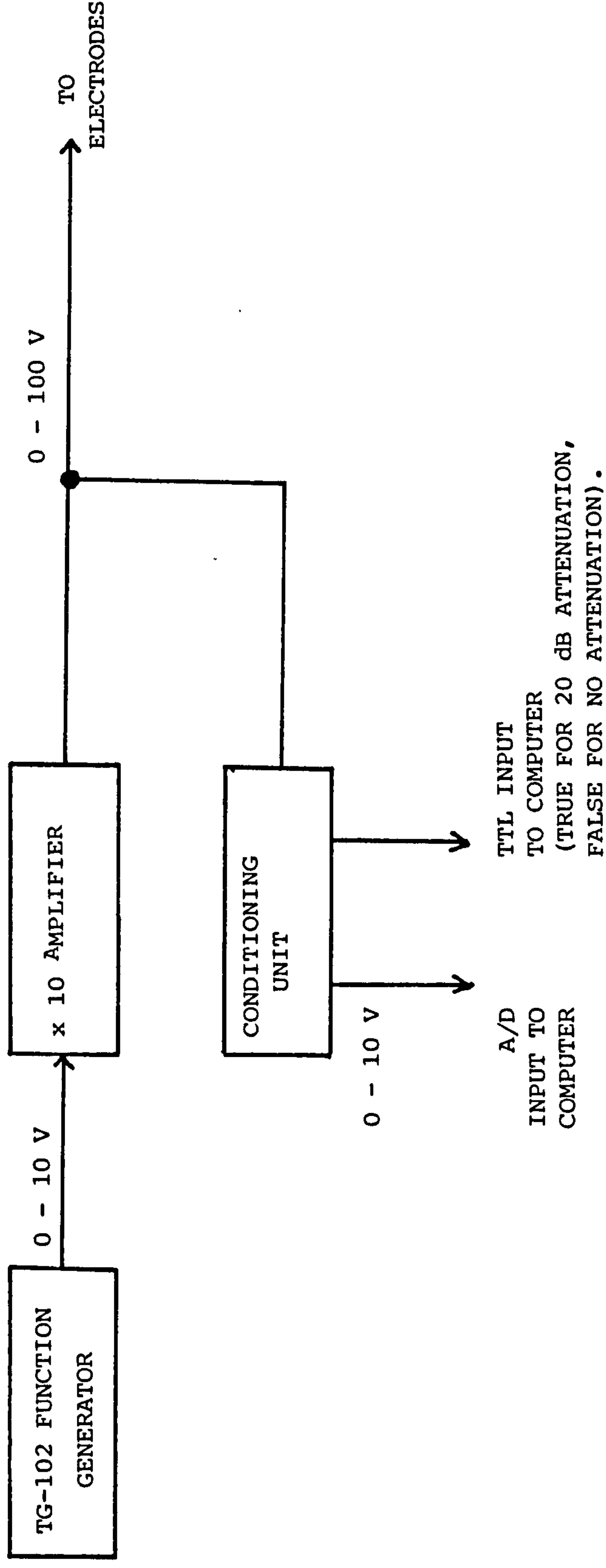


Figure 4.35: Schematic representation of how electrode signal conditioning unit is incorporated into the PALS apparatus.

following manner. For input amplitudes up to 10V, the input and output voltages were measured using the DT 2818 ADC board (Table 4.1). For input amplitudes above 10V, the input voltage was measured using the spectrum analyser (Table 4.2). Any systematic errors occurring between the spectrum analyser and the DT2818 were corrected for by using both pieces of equipment to determine the amplitude of a sinusoidal signal (Table 4.3). From these calibrations the following relationships were obtained for the input voltage,  $V_{IN}$ , and the output voltage  $V_{OUT}$ ,

$$V_{OUT} = (1.0230 \pm 0.0006) V_{IN} \text{ for } V_{IN} < 10 \text{ V} \quad (4.3)$$

and

$$V_{OUT} = (0.1064 \pm 0.0004) V_{IN} \text{ for } V_{IN} > 10 \text{ V} \quad (4.4)$$

#### 4.2.13 CONSTANT VELOCITY TERM MINIMISATION

In § 3.6.6, it was mentioned that by measuring the amplitude and phase information of the scattered light signal relative to a frequency  $\Delta\omega_s + qV_c$ , rather than  $\Delta\omega_s$ , the constant velocity term,  $qV_c$ , in the AWPD function ( $\pi/4 \cdot q^2 V_c^2$  for the AWPS function) could be removed, leaving only electrophoretic and diffusive information to be processed by the computer. The following scheme has been incorporated into the PALS equipment in order to minimise the constant velocity contribution to the demodulated real and imaginary signals. The

Input amplitude/V	Output amplitude/V
0.3369	0.3467
1.0938	1.1255
1.9873	2.0361
2.9907	3.0664
4.0845	4.1846
5.0488	5.1709
6.0425	6.1938
7.9834	8.1787
8.9258	9.1406
9.7363	9.9536

Table 4.1: Calibration data for electrode signal conditioning unit  
for input amplitudes up to 10V (i.e. no attenuation).

Input amplitude/V	Output amplitude/V
10.055	1.0522
12.714	1.3306
15.698	1.6650
19.799	2.0972
24.324	2.5879
29.698	3.1592
35.921	3.8086
42.144	4.5020

Table 4.2: Calibration data for electrode signal conditioning unit  
for input amplitudes over 10V (i.e. ~ 20 dB attenuation).

$V_{DT2818}$	$V_{SPEC}$	$V_{DT2818}$	$V_{SPEC}$
1.0229	1.0080	6.0620	6.0104
2.0020	1.9799	7.2070	7.1418
2.9395	2.8991	8.1738	8.1034
3.9624	3.8608	9.0820	8.9944
5.0342	4.9780		

Table 4.3: Correction data for voltages determined using DT2818 ( $V_{DT2818}$ ) and 3582A spectrum analyser ( $V_{SPEC}$ ).



5210 LIA's internal oscillator provides the LF input to the SSBM. However, a TG-102 function generator is used to generate an external reference signal for the 5210. One important feature of the TG-102 is that its output frequency can be adjusted by application of an external voltage ("sweep voltage"). The frequency change is linear with the applied voltage. During a PALS experiment, which will normally consist of, say, 5 runs of 500 averages, with each average containing 50 to 150 data points at 1 ms spacings, the software initially measures the demodulation frequency via the 5210 LIA. It then applies two known voltages (e.g.  $\pm 0.1\text{V}$ ), via one of the DT2818's D/A channels, to the TG-102's sweep input. For each of the voltages, the output frequency is determined. Thus, the TG-102 is calibrated in terms of the frequency change occurring for a given applied "sweep" voltage. Between each experimental run, the software determines the gradient of the AWPD function. If the gradient is  $\omega \text{ rad s}^{-1}$ , then the voltage applied to the TG-1202 is changed by the necessary amount to increase the output frequency by  $\omega \text{ rad s}^{-1}$ . In practice, it is not possible to completely remove the constant velocity term, but it is possible to maintain it below  $1 \mu\text{m s}^{-1}$ . This frequency tracking procedure is the "Autotrack" facility found in the software listing.

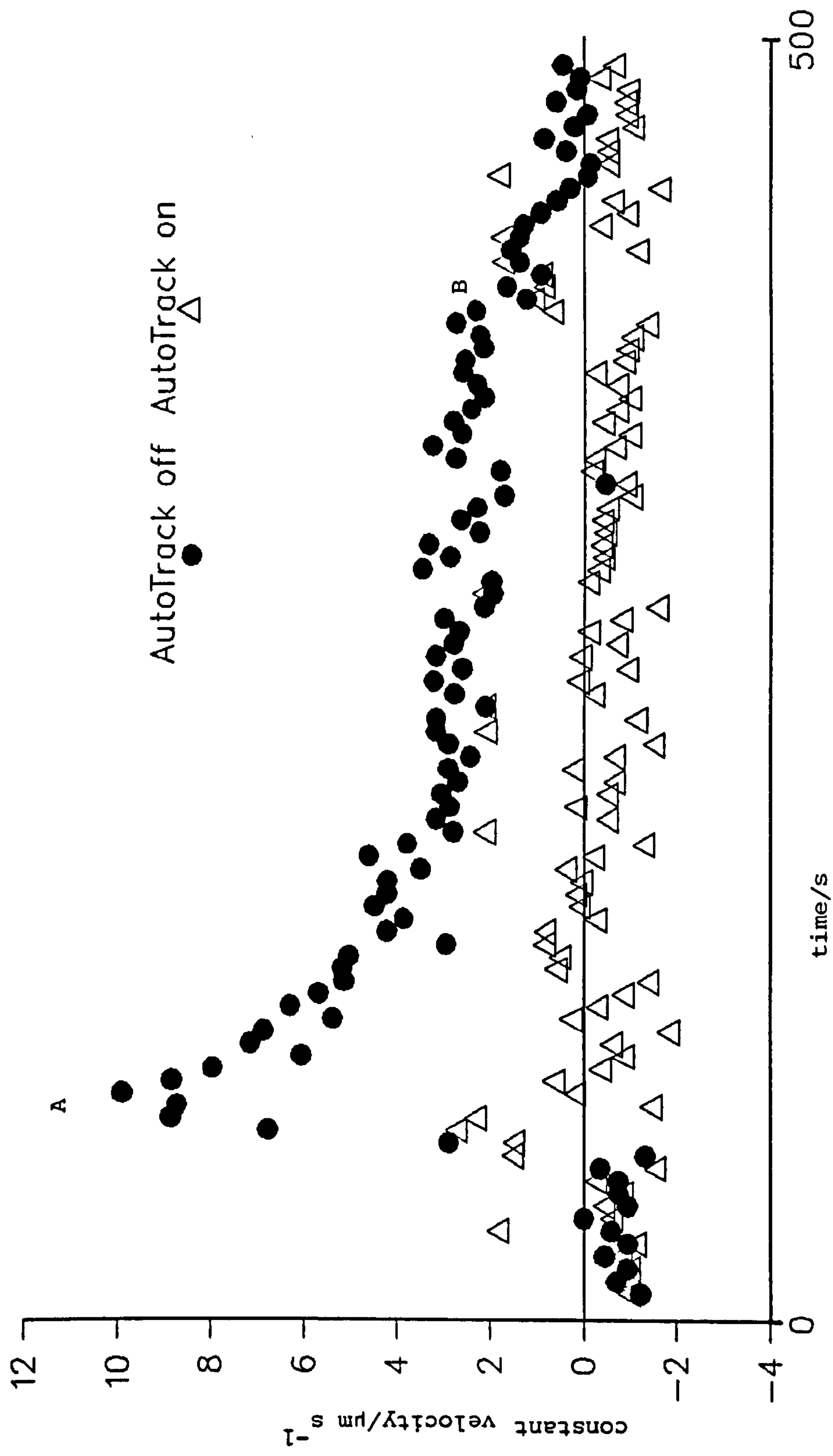
The effectiveness of this "Autotrack" feature can be seen in the following study. A dilute dispersion of  $\sim 350 \text{ nm}$  diameter silica particles in water was placed in the sample cell. The ambient temperature was  $22^\circ\text{C}$ . Water at  $25^\circ\text{C}$  was circulated through the thermostating jacket after a couple of minutes. The constant velocity



term for the sample was determined via the AWPD function without constant velocity minimisation enabled. The e<sup>x</sup>Periment was repeated but with the minimisation procedure operative. Figure 4.36 shows how the constant (i.e. convective) velocity varies as a function of time. Point A indicates when the circulating pump was turned on, pumping at its lowest flow rate. Point B indicates when the flow rate was increased. Each data point is obtained from an AWPD function consisting of 50 averages, each containing 50, 1 ms-spaced points. It is clearly evident, that, even for a continuously varying "constant" velocity term, the minimisation scheme is very effective. This is very important when wishing to study dispersions with very small electrophoretic mobilities.

#### 4.3 FINAL INSTRUMENT VALIDATION

The validation of PALS for determining electrophoretic mobilities was confirmed by studying a wide range of aqueous and non-aqueous dispersions. The mobilities were determined by PALS ( $\mu_{\text{PALS}}$ ) and conventional LDE ( $\mu_{\text{LDE}}$ ) using the same basic apparatus. Thus, it was only necessary to verify that PALS data processing yielded the same electrophoretic velocity information for a given dispersion as obtained by conventional spectral analysis. Any systematic errors arising from incorrect determination of the applied field strength or scattering vector would affect  $\mu_{\text{PALS}}$  and  $\mu_{\text{LDE}}$  to the same degree.



**Figure 4.36:** Graph to illustrate the effectiveness of constant velocity minimisation.

See text for details.

A comparison between mobilities measured using a commercial instrument (Pen Kem Inc. System 3000),  $\mu_{\text{PENKEM}}$ , and the PALS equipment was made to confirm that the PALS equipment yields accurate velocity information.

The effect of varying a number of experimental parameters on  $\mu_{\text{PALS}}$  was studied, with particular regard to the statistical uncertainty in the result. The results presented here illustrate the versatility of the PALS technique.

Tables 4.4(a) and 4.4(b) show the results for mobility measurements of a dilute dispersion of ~ 500 nm diameter polystyrene latex particles in  $10^{-3}$  mol dm $^{-3}$  aqueous potassium chloride solution at pH 10 as determined using PALS and conventional LDE with the same equipment. The excellent agreement between  $\mu_{\text{PALS}}$  ( $-(2.95 \pm 0.03) \times 10^{-8}$  m $^2$  s $^{-1}$  V $^{-1}$ ) and  $\mu_{\text{LDE}}$  ( $-(2.93 \pm 0.09) \times 10^{-8}$  m $^2$  s $^{-1}$  V $^{-1}$ ) confirms that the PALS processing technique does yield the same velocity information as obtained via conventional spectral analysis of the scattered light signal. Table 4.5 shows the mobility results for 6 typical aqueous dispersions measured by PALS and the commercial instrument mentioned earlier. A plot of  $\mu_{\text{PENKEM}}$  vs.  $\mu_{\text{PALS}}$  yields

$$\mu_{\text{PENKEM}} = (0.97 \pm 0.04) \mu_{\text{PALS}} + (0.02 \times 10^{-8} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1})$$

Electric field strength/kV m <sup>-1</sup>	Electric field frequency / Hz	Mobility, $\mu_{\text{LDE}}$ /10 <sup>-8</sup> m <sup>2</sup> s <sup>-1</sup> V <sup>-1</sup>
1.76	1.5	-3.16 ± 0.05
1.33	1.5	-3.09 ± 0.11
0.83	1.5	-2.67 ± 0.05
0.83	3.0	-2.83 ± 0.12
1.32	3.0	-2.93 ± 0.04
1.76	3.0	-2.89 ± 0.14

Table 4.4(a): Conventional LDE mobilities for ~ 500 nm diameter polystyrene latex particles in 10<sup>-3</sup> mol dm<sup>-3</sup> KCl(aq) at pH 10.

Electric field strength/kV m <sup>-1</sup>	Electric field frequency /Hz	Mobility, $\mu_{\text{PALS}}$ /10 <sup>-8</sup> m <sup>2</sup> s <sup>-1</sup> V <sup>-1</sup>
0.36	15.0	-2.83 ± 0.06
0.73	15.0	-2.97 ± 0.01
1.10	15.0	-2.98 ± 0.01
1.57	15.0	-2.97 ± 0.02
0.34	30.0	-2.90 ± 0.06
0.76	30.0	-3.00 ± 0.03
1.37	30.0	-2.99 ± 0.02
2.08	30.0	-2.92 ± 0.01

Table 4.4(b): PALS mobilities for ~ 500 nm diameter polystyrene latex particles in 10<sup>-3</sup> mol dm<sup>-3</sup> KCl(aq) at pH 10.

SAMPLE	$\mu_{\text{PALS}}$ / $10^{-8} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$	$\mu_{\text{PENKEM}}$ / $10^{-8} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$
A	-2.36	-2.29
B	-5.43	-5.68
C	-4.85	-4.30
D	-0.99	-1.08
E	+1.56	+1.68
F	+2.57	+2.54

Table 4.5: Comparison between  $\mu_{\text{PALS}}$  and  $\mu_{\text{PENKEM}}$  for six aqueous samples :

- A: ~ 630 nm diam. Polystyrene latex/0.002  
mol dm<sup>-3</sup> KCl at pH 5.7
- B: ~ 630 nm diam. polystyrene latex/0.0025  
mol dm<sup>-3</sup> phosphate buffer at pH 7.0
- C: ~ 670 nm diam. silica particles/0.025  
mol dm<sup>-3</sup> phosphate buffer at pH 7.0
- D: ~ 670 nm diam. silica particles/0.0025  
mol dm<sup>-3</sup> biphthalate buffer at pH 4.0
- E: ~ 70 nm diam. haematite particles/0.002  
mol dm<sup>-3</sup> KCl at pH 4.0
- F: ~ 70 nm diam. haematite particles/0.0005  
mol dm<sup>-3</sup> cetyl trimethylammonium bromide/  
0.0025 mol dm<sup>-3</sup> phosphate buffer at pH 7.0

(These samples were supplied for measurement by A. van der Linde, Wageningen Agricultural University, The Netherlands).



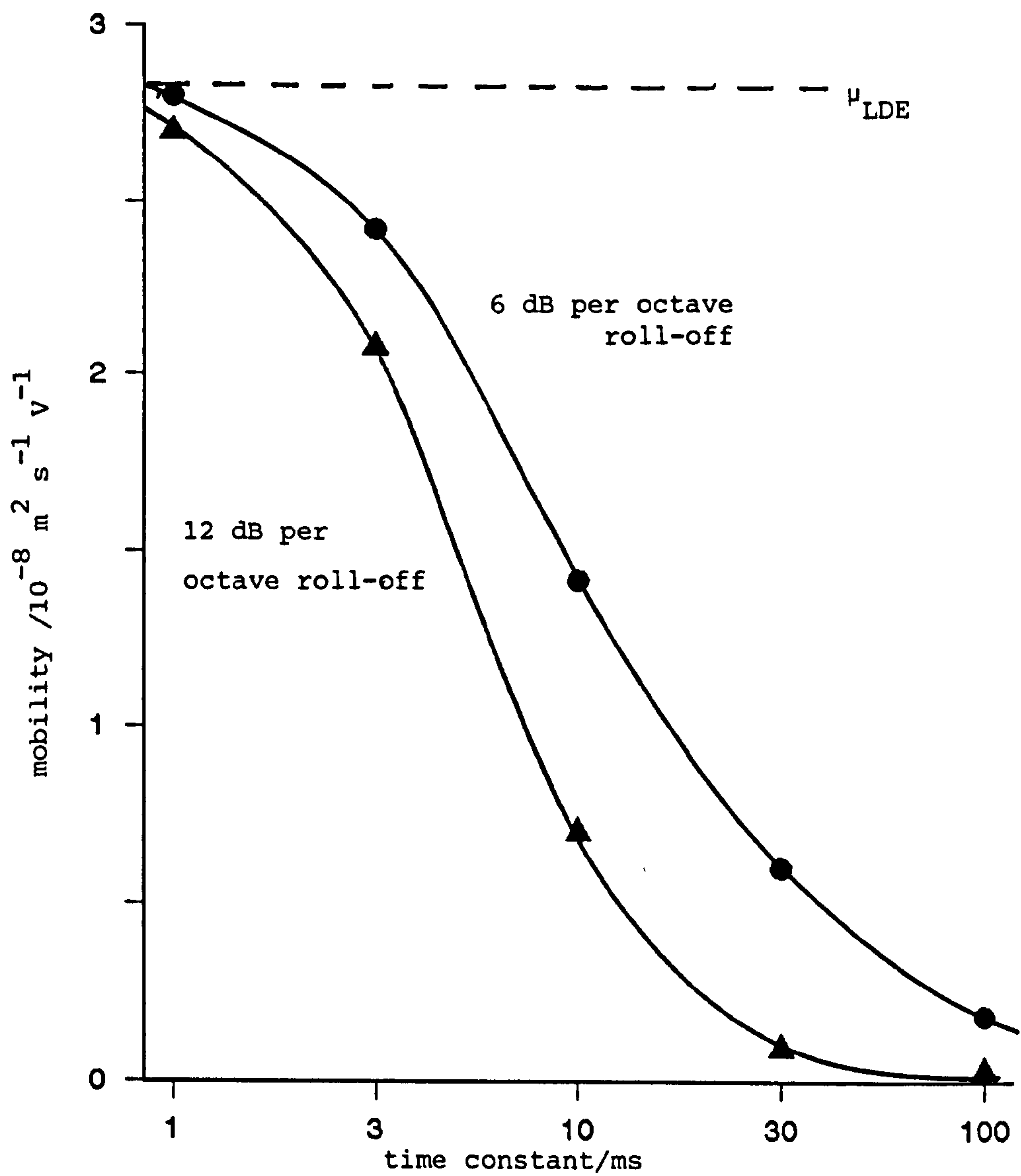
with a correlation coefficient of 0.997. This confirms the validity of PALS for the accurate determination of electrophoretic mobilities. It is now necessary to consider how variable experimental parameters, such as fringe frequency, applied field strength and frequency, affect both the systematic and statistical uncertainty of  $\mu_{\text{PALS}}$ .

Table 4.6 shows how the fringe frequency influences the determined electrophoretic mobility of  $\sim 500$  nm diameter polystyrene latex particles in  $10^{-3}$  mol dm $^{-3}$  aqueous potassium chloride. The decrease in  $\mu_{\text{PALS}}$  below 1 kHz and above 2.4 kHz is due to the limited bandwidth of the SSBM. A fringe frequency of 1.95 to 2.00 kHz is usually employed.

Figure 4.37 shows how  $\mu_{\text{PALS}}$  for the same dispersion varies as a function of the 5210 LIA's post-detector smoothing time constant, confirming the importance of using minimum output smoothing.

The electrophoretic mobility of an aqueous dispersion of  $\sim 430$  nm diameter cross-linked polystyrene latex particles in  $10^{-3}$  mol dm $^{-3}$  aqueous potassium chloride solution was determined as a function of applied field strength and the number of averages per experiment, at a fixed field frequency of 30.00 Hz. In each case, five experiments were performed to estimate the uncertainty in  $\mu_{\text{PALS}}$ . Table 4.7 summarises these results. As expected, the uncertainty





**Figure 4.37:** Electrophoretic mobility (via PALS) for ~ 500 nm diameter polystyrene latex in  $10^{-3} \text{ mol dm}^{-3}$  KCl (aq) as a function of lock-in amplifier filter time constant.

Fringe frequency /Hz	$\mu_{\text{PALS}} / 10^{-8} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$
250	$-2.20 \pm 0.03$
501	$-2.55 \pm 0.02$
1000	$-2.67 \pm 0.03$
1497	$-2.76 \pm 0.03$
2000	$-2.76 \pm 0.01$
2100	$-2.76 \pm 0.01$
2200	$-2.73 \pm 0.03$
2300	$-2.77 \pm 0.03$
2400	$-2.74 \pm 0.04$
2480	$-2.39 \pm 0.05$

Table 4.6: Influence of fringe frequency on  $\mu_{\text{PALS}}$  for  
~ 500 nm diam. Polystyrenelatex particles in dilute  
KCl (aq).

No. of	Applied field/kV m <sup>-1</sup>				
Averages	0.34	0.53	1.05	1.58	2.11
50	12.0	-	3.46	2.69	2.75
100	-	4.05	3.82	1.34	0.70
250	-	2.81	1.71	1.03	1.72
500	-	1.41	0.69	1.03	1.36
1000	-	1.36	0.64	0.32	0.96

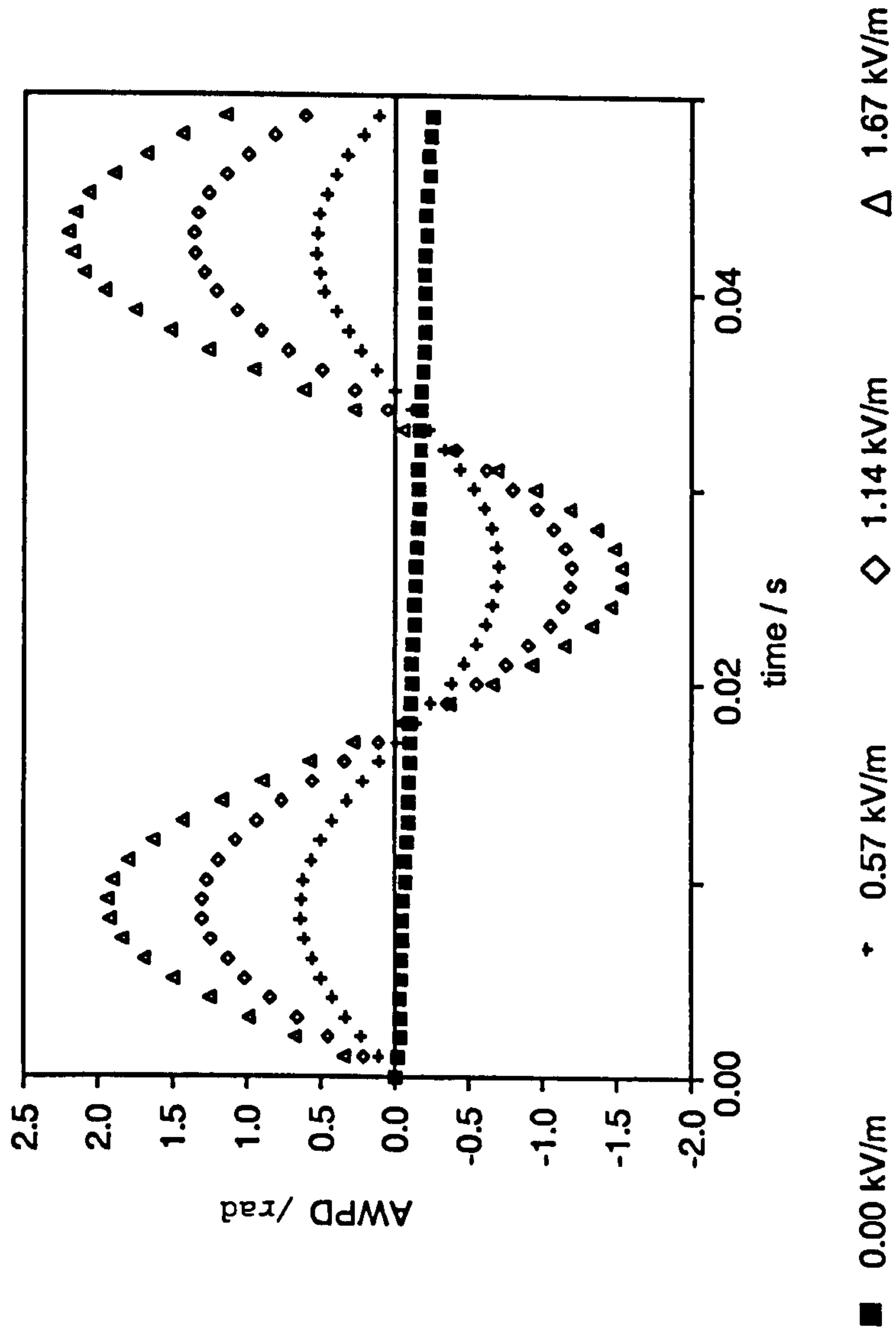
Table 4.7: Statistical uncertainty in  $\mu_{\text{PALS}}$ , expressed as a percentage standard deviation, for ~ 430 nm diam. cross-linked polystyrene latex particles in 0.001 mol dm<sup>-3</sup> KCl (aq) as a function of applied field strength and number of averages at a fixed field frequency of 30.0Hz.

in the mobility decreases both with increasing field strength and the number of averages per experiment. In each experiment, each average consisted of 50 points spaced at 0.999 ms intervals. Thus, the total experimental duration is  $(5 \times N \times 50 \times 0.999 \times 10^{-3})\text{s}$ , where  $N$  is the number of averages indicated in table 4.7. (See Appendix 2 for an explanation of how the PALS software achieves realtime data capture and processing). It has been found from experience that 500 averages are sufficient to yield uncertainties of the order of 1% for samples with peak electrophoretic velocities of  $\sim 5 \mu\text{m s}^{-1}$ . The effect of varying the field frequency on  $\mu_{\text{PALS}}$  for the same system is summarised in table 4.8. As can be seen, within experimental uncertainty, there is no field frequency dependence up to at least 45 Hz. It is worth noting at this point that this is in direct contrast with the claims of Gaigalas et al (see § 3.6.7). The AWPDP function provides a direct measurement of the first moment of sinusoidal phase changes occurring due to light scattered by particles moving in a sinusoidal electric field, and it is shown here that the above workers' claims are incorrect.

Figure 4.38 shows typical AWPDP functions for an aqueous dispersion of  $\sim 500$  nm diameter polystyrene latex particles. The agreement between the experimental data and theory in each case is excellent. Each of the functions shown represent 25 seconds of processed scattered light signal.

Frequency/Hz	Field/kV m <sup>-1</sup>	$\mu_{\text{PALS}} / 10^{-8} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$
10.0	0.32	-2.86 ± 0.10
10.0	1.04	-2.97 ± 0.05
15.0	0.36	-2.83 ± 0.06
15.0	0.73	-2.97 ± 0.01
15.0	1.10	-2.98 ± 0.01
15.0	1.57	-2.97 ± 0.02
30.0	0.34	-2.90 ± 0.06
30.0	0.76	-3.00 ± 0.03
30.0	1.37	-2.99 ± 0.02
30.0	2.08	-2.92 ± 0.01
45.0	0.56	-2.87 ± 0.03
45.0	1.06	-2.87 ± 0.03
60.0	1.10	-2.80 ± 0.07
60.0	2.07	-2.76 ± 0.04

Table 4.8:  $\mu_{\text{PALS}}$  for ~ 430 nm diam. cross-linked polystyrene latex particles in 0.001 mol dm<sup>-3</sup> KCl (aq) as a function of applied field frequency and strength.



**Figure 4.38:** AWPD functions for ~ 500 nm diameter polystyrene latex particles in aqueous KCl solution at various applied field strengths.



The electrophoretic mobilities given in table 4.9 for an aqueous dispersion of ~ 630 nm diameter cross-linked polystyrene particles show that measured mobilities are independent of applied field strength and scattering angle. It is interesting to note that each of the mobility values in table 4.9 arose from analysis of  $5 \times 50 \times 50 \times 0.5 \times 10^{-3} = 6.25\text{s}$  of data. Even though this time is relatively short, uncertainties of the order of 2% are achieved.

From the above sets of results, it is clear that PALS provides a very quick and statistically accurate means of measuring electrophoretic mobilities of aqueous colloidal dispersions. It is now necessary to consider the determination of very small electrophoretic mobilities of non-polar colloidal dispersions. For this purpose, ~ 1.1  $\mu\text{m}$  diameter cross-linked polystyrene latex particles ("PSLX-PP-1", see chapter 5 for preparative details) in toluene and 1, 4-dioxan were used.

Figure 4.39 (a) shows the AWPDP and AWPS functions for PSLX-PP-1 dispersed in 1, 4-dioxan with no applied field. The sampling was 2500 samples of 50 1 ms-spaced data points (i.e.  $2500 \times 50 \times 10^{-3} \text{ s}$ ), and the scattering angle was  $15.2^\circ$  in air. Figure 4.39 (b) shows the AWPDP and AWPS functions for the same sample but with an applied sinusoidal field of  $100 \text{ kV m}^{-1}$  at 30.00 Hz. As can be seen, a constant velocity

Scattering	Field strength					
Angle / deg	(V across 1.56 mm)					
	2	3	4	6	8	10
12.93	-2.21	-2.23	-2.20	-2.23	-2.33	-2.28
	±0.08	±0.07	±0.04	±0.07	±0.01	±0.05
14.74	-2.25	-2.28	-2.33	-2.29	-2.38	-2.32
	±0.05	±0.03	±0.09	±0.06	±0.07	±0.08
21.93	-2.16	-2.29	-2.25	-2.28	-2.32	-2.34
	±0.10	±0.04	±0.03	±0.04	±0.02	±0.05

Table 4.9:      Variation of  $\mu_{\text{PALS}}$  for a dilute aqueous dispersion of ~ 630 nm diam. cross-linked polystyrene particles as a function of field strength and scattering angle. Sampling = 5 x 50 x 50 x 0.5 ms, field frequency = 45 Hz. Mobilities quoted in  $10^{-8} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$ .

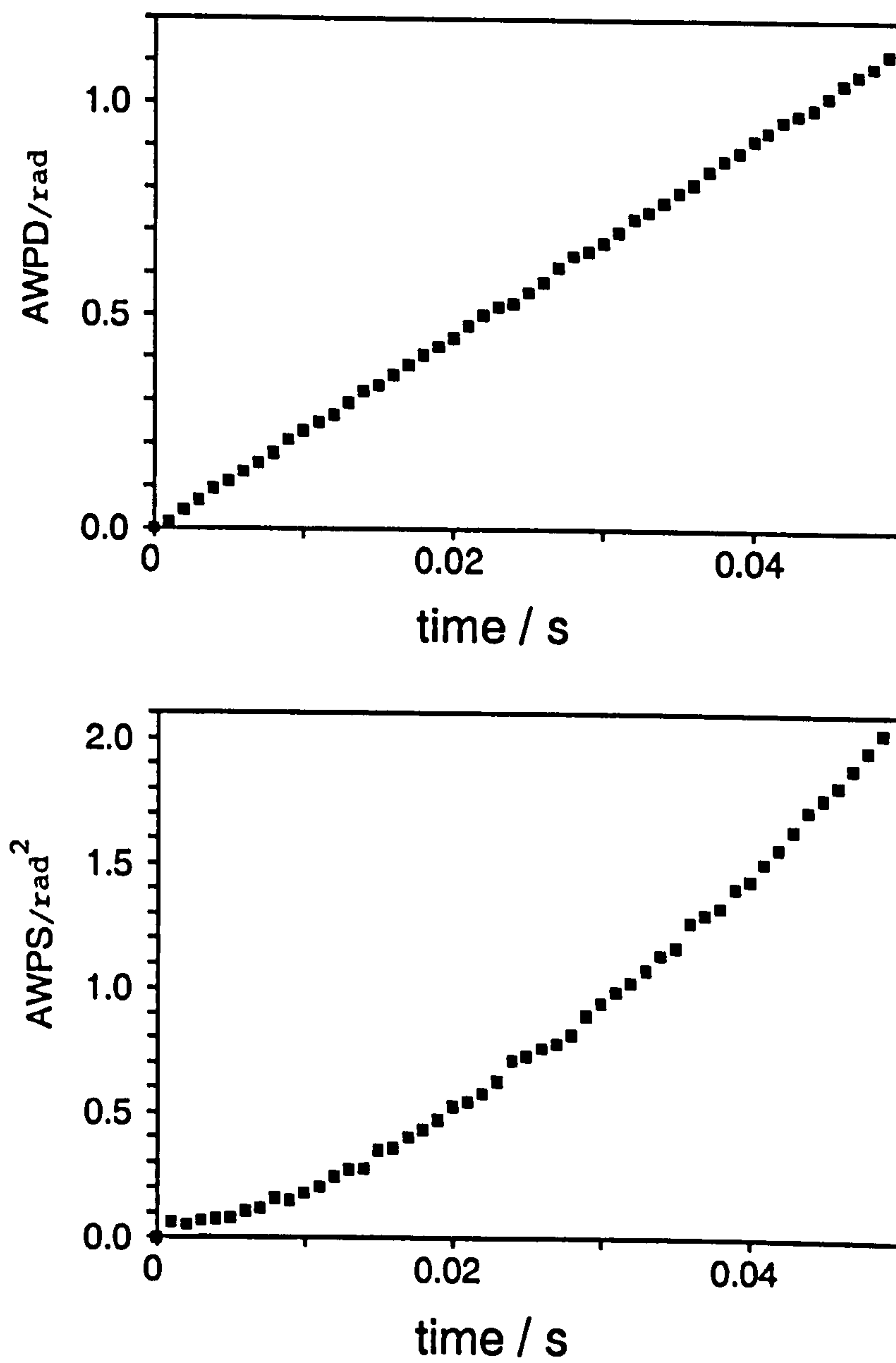


Figure 4.39(a): AWP and AWPS functions for PSLX-PP-1 dispersed in 1,4-dioxan in the absence of an electric field.

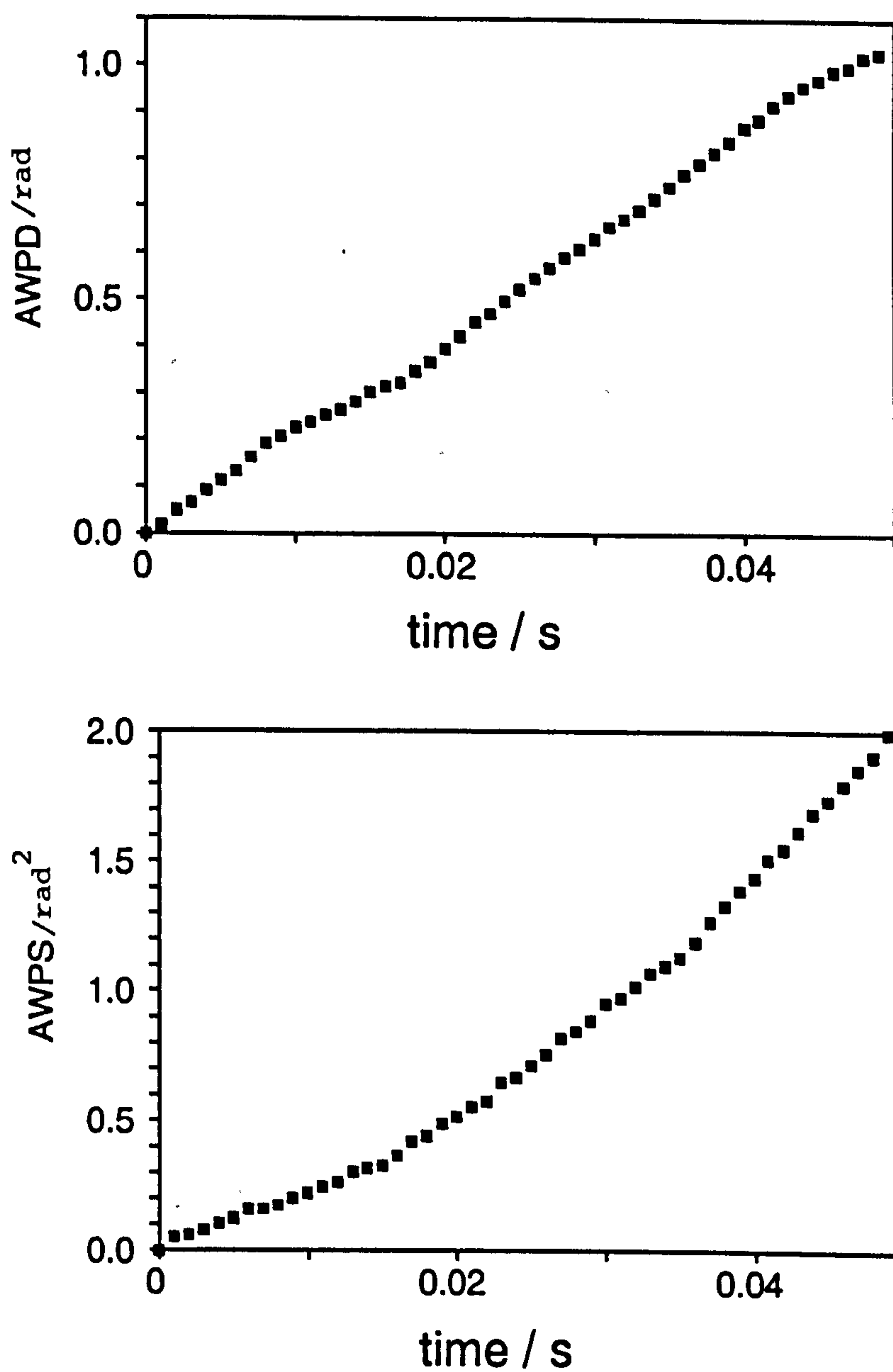


Figure 4.39(b): AWPDP and AWPS functions for PSLX-PP-1 dispersed in 1,4-dioxan subjected to an applied field of  $100 \text{ kV m}^{-1}$  at 30.0 Hz, with phase demodulation occurring at fringe frequency.

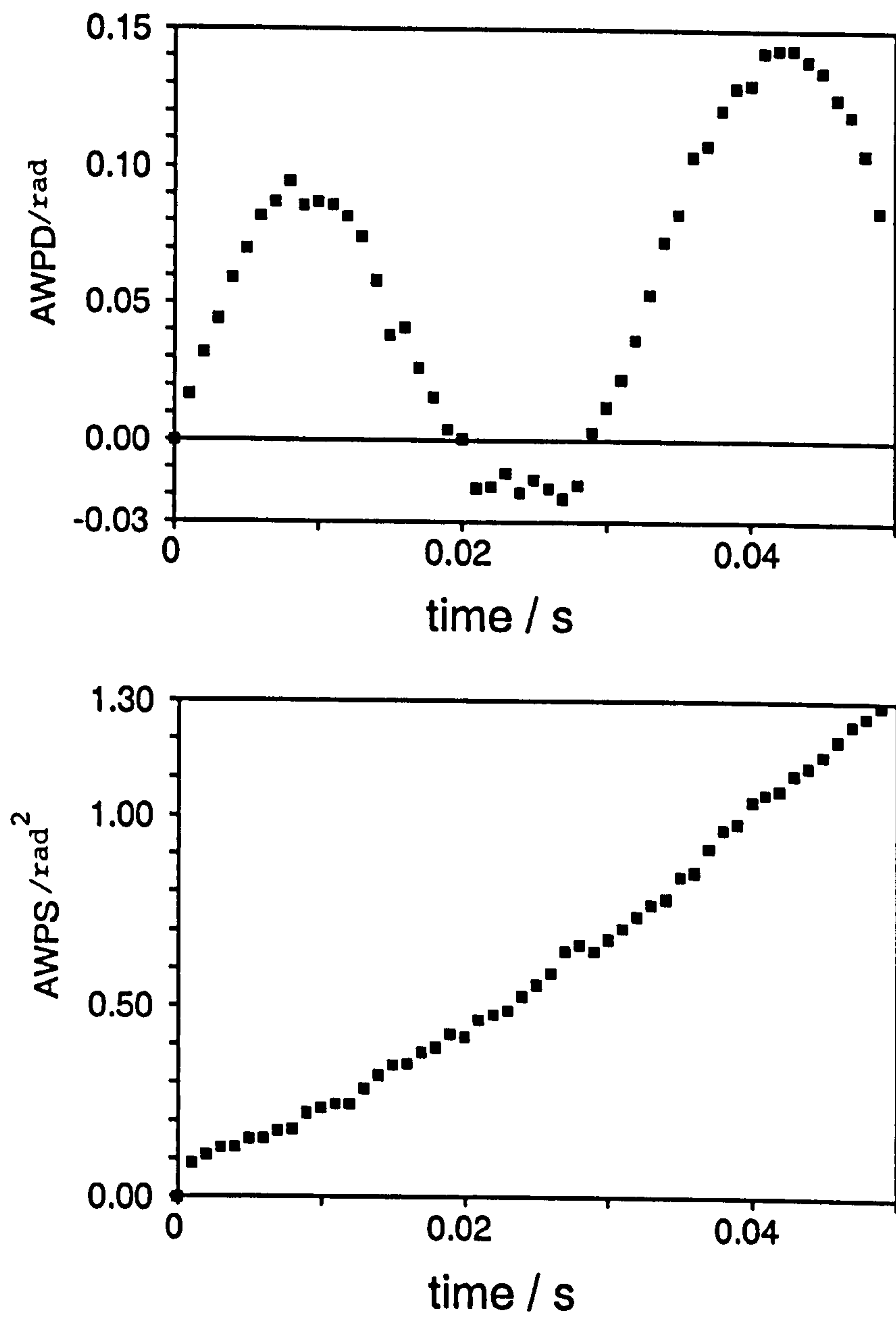


Figure 4.39(c): As for fig 4.39(b), but with demodulation at a frequency chosen to minimise constant velocity component in AWPD/AWPS functions.

term dominates both functions, making reliable determination of the mobility difficult. However, figure 4.39 (c) shows the AWPD and AWPS functions for the same system subjected to the same electric field, but with the determination of the amplitude and phase data relative to a frequency suitable to minimise the constant velocity term, as described in § 4.2.13. The sinusoidal term in the AWPD is now more dominant, increasing the reliability of the mobility determination. The AWPS function, however, is dominated by diffusion and shows very little sinusoidal component. This highlights the need to use the AWPD function, rather than the AWPS function, if very small mobilities are to be determined. The AWPS function is useful, though, allowing the diameter of the dispersed particles to be estimated. The sample used here has a diameter of  $1184 \pm 41$  nm in 1, 4-dioxan at  $25^{\circ}\text{C}$ , as determined by P.C.S. (see next chapter). The average of 5 PALS experiments (total duration 10 min 25 sec) yielded a particle diameter of  $1181 \pm 74$  nm for the same sample.

Autocorrelation of the real component of the demodulated scattered light signal also enables estimation of the particle size. The size of the above particles determined this way was  $1201 \pm 168$  nm. Figure 4.40 shows the normalised autocorrelation function for the PSLX-PP-1 / dioxan dispersion, obtained in real time simultaneously with the above - mentioned AWPD and AWPS functions. For this dispersion, the electrophoretic mobility of  $-(4.31 \pm 0.14) \times 10^{-11} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$  corresponds to a displacement of  $\sim 60$  nm per electrode cycle. In a



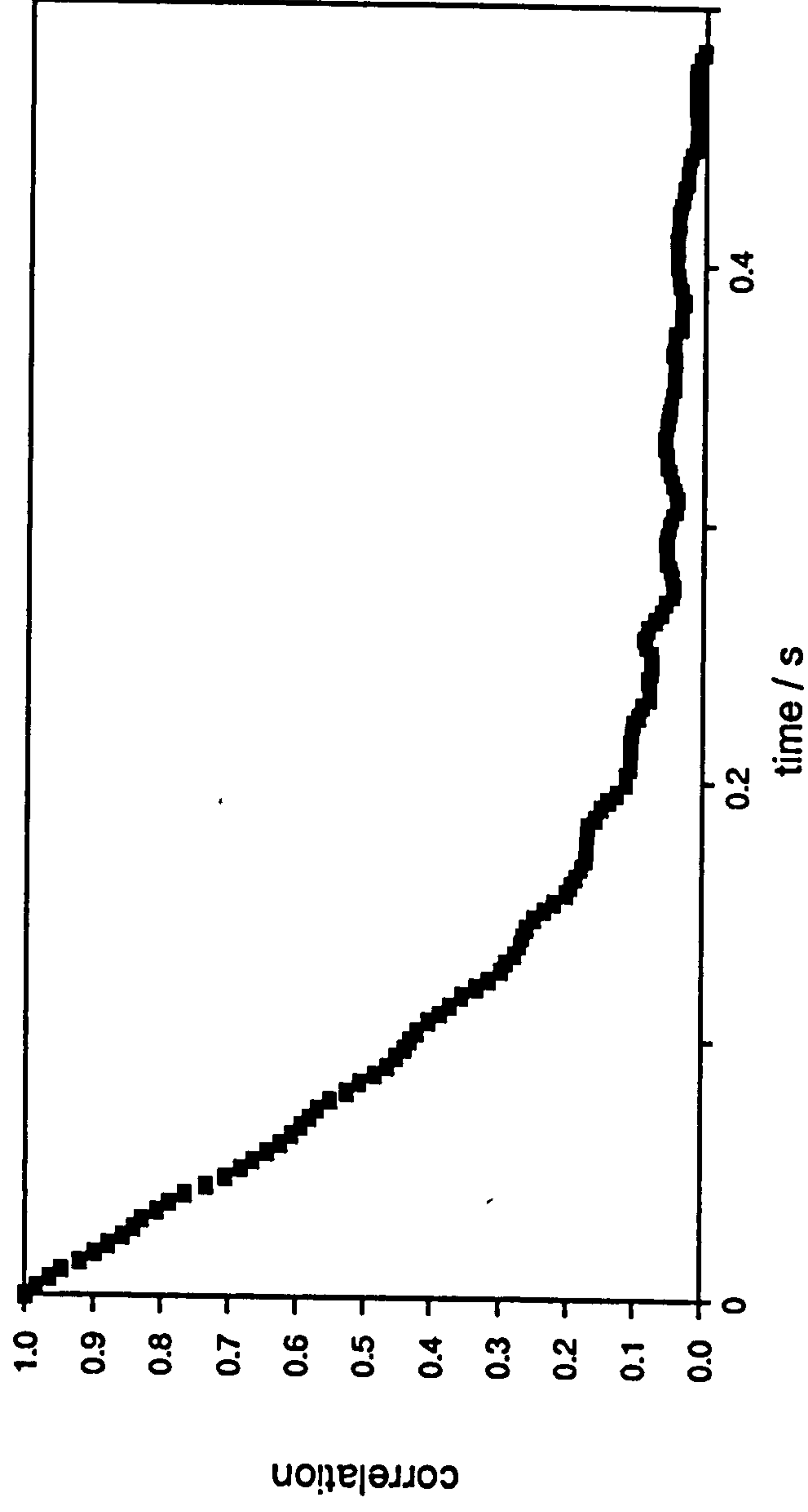


Figure 4.40: Autocorrelation function of the real component of the demodulated scattered light signal for PSLX-PP-1 in 1,4-dioxan. Yields particle diameter of 1165nm.

d.c. field of  $100 \text{ kV m}^{-1}$ , the same particles would yield a classical Doppler shift of  $- 3.5 \text{ Hz}$ , which is less than the spectral broadening due to diffusion ( $\sim 16 \text{ Hz}$ ). Thus, classical LDE would not be able to resolve, the electrophoretic motion from this type of sample at the quoted field strength.

Figure 4.41 shows the amplitude histogram obtained during the experiment described for figure 4.39 (c). The smoothness of this histogram can be used to inform the experimenter that sufficient samples have been averaged to ensure statistically reliable results. Any extra maxima occurring at high amplitudes serve to indicate dust contamination of the sample, although it has been found that this is rarely a problem. When dust is present, most data arising from the scattering due to dust is rejected (§ 3.6.9 and Appendix 2).

The usefulness of measuring very small electrophoretic mobilities in non-polar dispersions can be seen in the following results. Silica particles ( $\sim 100 \text{ nm}$  diameter) with an adsorbed layer of an AB-block copolymer (poly(dimethylaminoethyl methacrylate)-co-poly(methyl methacrylate)), to confer stability in 1, 4-dioxan<sup>117</sup>, was studied to verify that electrostatic repulsions have negligible influence on the kinetic stability of the dispersion. The electrophoretic mobility was found to be  $+ (4.79 \pm 0.56) \times 10^{-11} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$ , and the diameter

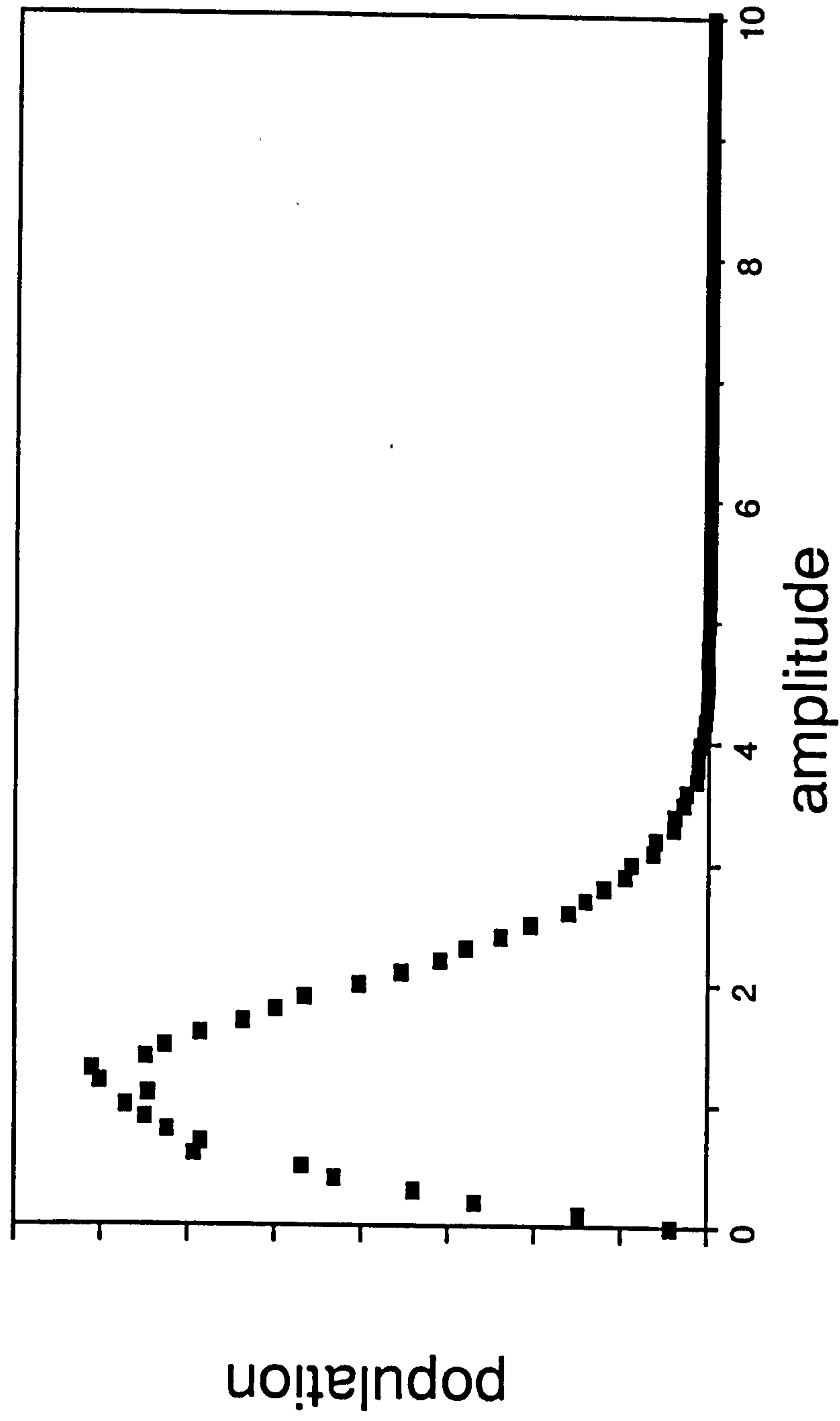


Figure 4.41: Amplitude histogram for PSLX-PP-1 in 1,4-dioxan.

$120 \pm 8$  nm. Table 4.10 shows the results for the four individual experiments used to obtain the above data. The equivalent electrophoretic surface charge density,  $Q$ , obtained from  $\mu = Q/6\pi\eta a$ , where  $a$  = particle radius and  $\eta$  = solvent viscosity, is calculated to be  $-(0.39 \pm 0.05)$  electron charges per particle, and the zeta potential, using eq (2.43) is 2.5 mV. This confirms that electrostatic repulsive forces are negligible compared with steric forces, contributing less than 0.01 kT to the potential energy barrier between two approaching particles (eq (2.13) for  $H = 0$ ). Between each individual experiment, referred to in table 4.10, the apparatus automatically adjusted the demodulation frequency in order to minimise the constant velocity term.

Table 4.11 shows the electrophoretic mobility of  $\sim 1.5$   $\mu\text{m}$  diameter cross-linked polystyrene latex particles dispersed in toluene (0.01% solids by mass) as a function of the applied field strength. This confirms that the mobility is independent of the applied field strength.

Electrophoretic mobility / $10^{-11} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$	Constant velocity / $\mu\text{m s}^{-1}$	Diameter / nm
5.21	12.67	110.0
5.45	4.72	121.5
4.08	-4.23	131.2
4.41	0.79	118.1

Table 4.10: PALS results for silica/poly (dimethylaminoethyl methacrylate)-co- poly(methyl methacrylate) in 1,4-dioxan. Field = 98.22 V/1.00 mm at 30.0 Hz, sampling = 1000 x 100 x 0.999 ms.

Field/kV $\text{m}^{-1}$	$\mu_{\text{PALS}} / 10^{-10} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$
2.34	-4.35 $\pm$ 1.14
5.00	-4.42 $\pm$ 0.73
8.09	-4.53 $\pm$ 0.21
10.91	-4.62 $\pm$ 0.29
18.14	-4.83 $\pm$ 0.20
25.86	-4.40 $\pm$ 0.19
35.64	-4.28 $\pm$ 0.05

Table 4.11: Variation of  $\mu_{\text{PALS}}$  for cross-linked polystyrene latex particles in toluene with applied field strength.

## CHAPTER FIVE

### ELECTROPHORETIC STUDIES OF VARIOUS NON-AQUEOUS DISPERSIONS USING PALS

#### 5.1 GENERAL METHODS

##### 5.1.1 PARTICLE SIZE DETERMINATION

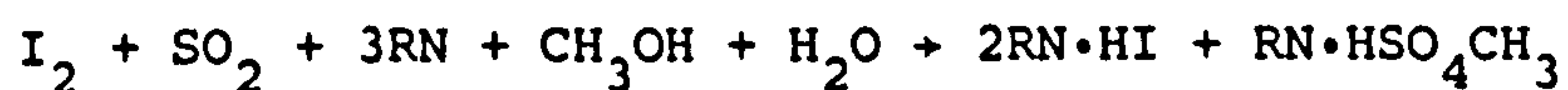
Particle size estimation for the dispersions described herein were performed using PCS (see § 3.3). All measurements were performed at 25°C. A coherent Super Graphite 3000K Krypton Ion laser was used to provide a 530.9 nm monochromatic light source, with a power between 0.2W and 1.0W. A Malvern Instruments PCS 100 spectrometer system was used for the sample chamber and light detecting hardware. The scattered light signal was processed by a Malvern Type K7027 digital autocorrelator. In all cases, diffusion coefficients were estimated by measuring the correlation function time constant,  $Dq^2$ , as a function of  $q^2$ . A linear plot of  $Dq^2$  versus  $q^2$  yields the desired diffusion coefficient,  $D$ .

Viscosity and relative permittivity values were obtained from the literature<sup>118, 119</sup>. Refractive index values were obtained from the literature<sup>119</sup> and by using an Abbe refractometer model 586269 (Bellingham and Stanley Limited). (Refractive index values are not required for PALS experiments).



### 5.1.2 WATER CONCENTRATION DETERMINATION

Water concentration determinations were performed using a Metrohm 652 (series 01) Karl-Fischer (KF) Coulometer. The principle of operation of the 652 KF device is a titration of water against iodine, in the presence of sulphur dioxide and an organic base, in methanol,



The Metrohm 652 performs the titration using controlled-current potentiometry with two indicator electrodes<sup>120</sup>, and can detect small amounts of water in the range 10 µg to 10 mg with a precision of 0.1 µg.

In most cases, approximately 250mg of sample (accurately weighed using a Precisa 80A 4-digit electronic balance) were injected into the titration vessel using a 2ml glass syringe and fine-bore, 3" stainless steel needle. For each sample, the water content was determined at least 3 times, until a constant reading was obtained. The syringe and needle were dried in an oven (70°C) before use.

### 5.1.3 PARTICLE MASS FRACTION DETERMINATION

The determination of the mass fractions of particles in dispersions was performed gravimetrically. Typically, 1 to 2ml of the sample were placed into a suitable pre-weighed container (e.g. a small glass jar or lid), and the total mass recorded. The solvent was evaporated, using an oven ( $70^{\circ} - 90^{\circ}\text{C}$ ), until the mass of the residue and container became constant. All weighings were carried out using a 4-digit electronic balance. For a given sample, typically 5 gravimetric analyses were performed.

### 5.1.4 CLEANING OF GLASSWARE

All glassware used throughout this work was cleaned by soaking overnight in aqua regia (3:1 by volume conc. nitric acid to conc. hydrochloric acid) or concentrated aqueous sodium hydroxide solution. After soaking, the glassware was thoroughly rinsed with water (ex. Millipore Milli-Q Water Purification System) and then steamed to remove any residual acidic or alkaline material. Sample cells, for PCS and PALS, were thoroughly dried overnight in a hot oven ( $90^{\circ}\text{C} - 100^{\circ}\text{C}$ ), and then "primed" with the relevant solvent before use. Where appropriate, glassware was subjected to ultrasonic treatment to aid cleansing.

#### 5.1.5 SOLVENT DRYING PROCEDURE

The following procedure was developed for drying solvents in preference to the more usual methods, such as addition of calcium hydride or metallic sodium.

Molecular sieve ( $3\text{\AA}$  diameter pore size, 8-12 mesh) was baked at  $300^{\circ}\text{C}$  for two hours. When having cooled sufficiently, it was placed in an 18" long, 1.5" diameter dried glass column, containing a glass sinter and tap, so as to provide a 12" long column of sieve. The column was used in the manner indicated in figure 5.1. The solvent to be dried, was passed through the column into the predried reception vessel, which also contained a small quantity of dry sieve. The 5ml syringe barrel (with sieve) and needle served as a pressure release outlet. Using this method, upto a litre of solvent could be dried within one hour. For example, one litre of toluene was dried from 150 ppm water by mass to 17 ppm by mass using the above method once. It was hoped that vacuum distillation of the dried solvent would remove water. However, it was found that, in the case of the toluene above, the water content increased again.

#### 5.1.6 DUST REMOVAL

For all light scattering experiments (i.e. PALS and PCS), dust was removed from solvents by filtering them through Sartorius

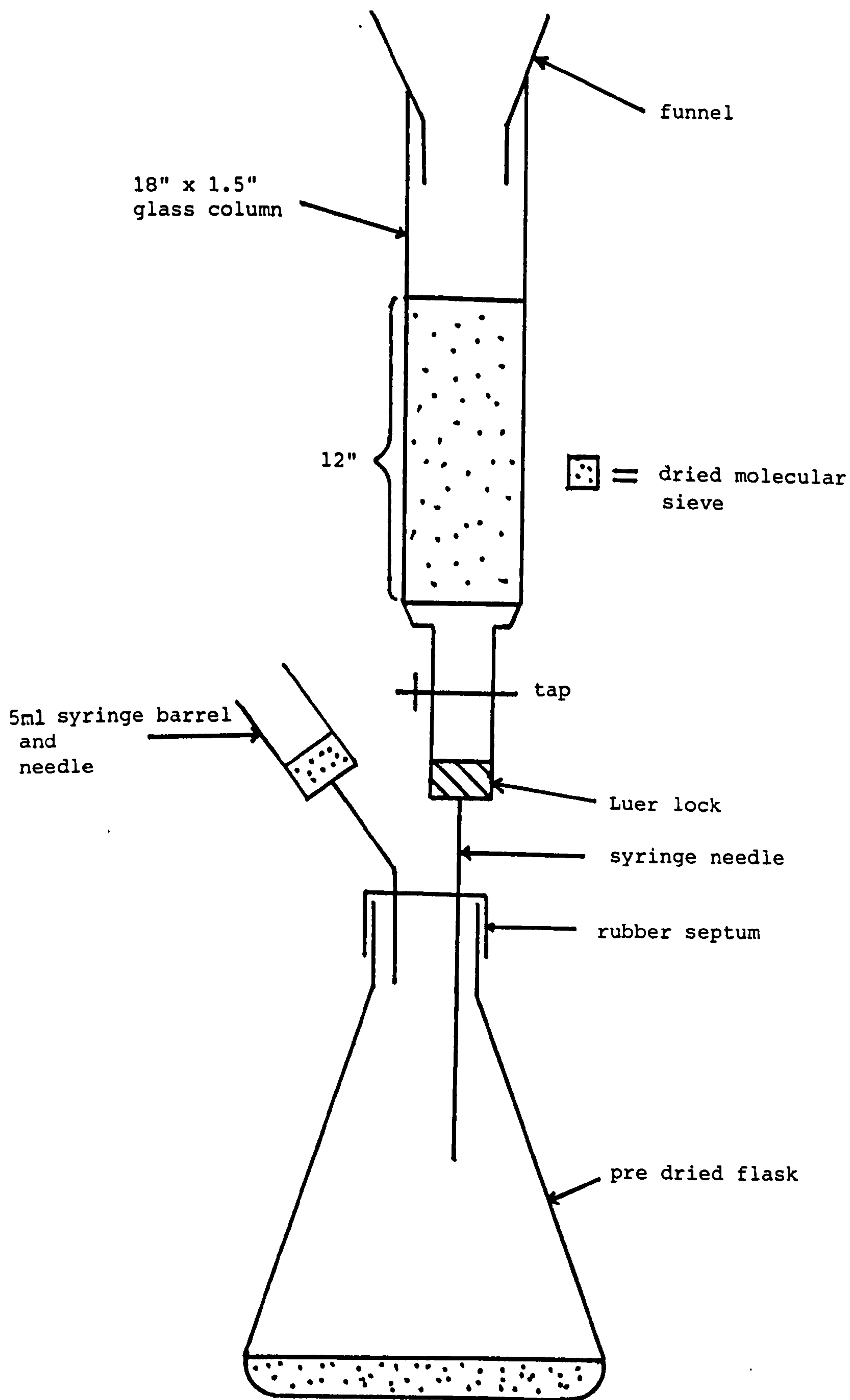


Figure 5.1: Solvent drying apparatus



type SM114 0.8 $\mu$ m cellulose nitrate membrane filters.

As has been mentioned before, PALS experiments can, to some degree, reject data caused by dust, and the amplitude histograms generated during PALS experiments can inform the user about the presence or absence of dust.

For PCS experiments, data that is likely to have been generated by scattering from dust can also be rejected. The K7027 correlator can determine the correlation function's baseline in two ways, either by a Swinbourne analysis or by direct measurement of the value of the decay at large delay times. For data free from dust contamination, the two values for the baseline should be very close (within 1%). However, contaminated data yield different baselines. Data are accepted or rejected by comparing the two values for the baseline.

#### 5.1.7 PREPARATION OF SAMPLES FOR MEASUREMENT

In studying the effects of small amounts of water on the electrophoretic properties of non-aqueous dispersions, it is very important that the samples are prepared in a low humidity environment, with as little contact with the open atmosphere as possible. Ideally, a high quality dry box should be used. Unfortunately, this was not possible and, hence, the following method was devised.

25ml glass bottles were used as sample tubes, with B14 size "Suba Seal" rubber septa used to seal the bottles. The bottles and septa were dried in an oven overnight and the septa inserted into the necks of the bottles, whilst hot. The required quantities of solvent and dispersion were injected into the bottles using dried, 2ml glass syringes with fine-bore needles. An extra needle was inserted into the septum in order to equalise the pressure inside and outside the bottles, whilst liquids were added. The same size syringes were used for removing small quantities of the samples for PALS, PCS and KF measurements.

## 5.2 MICROGEL LATEX PREPARATION

### 5.2.1 GENERAL DESCRIPTION

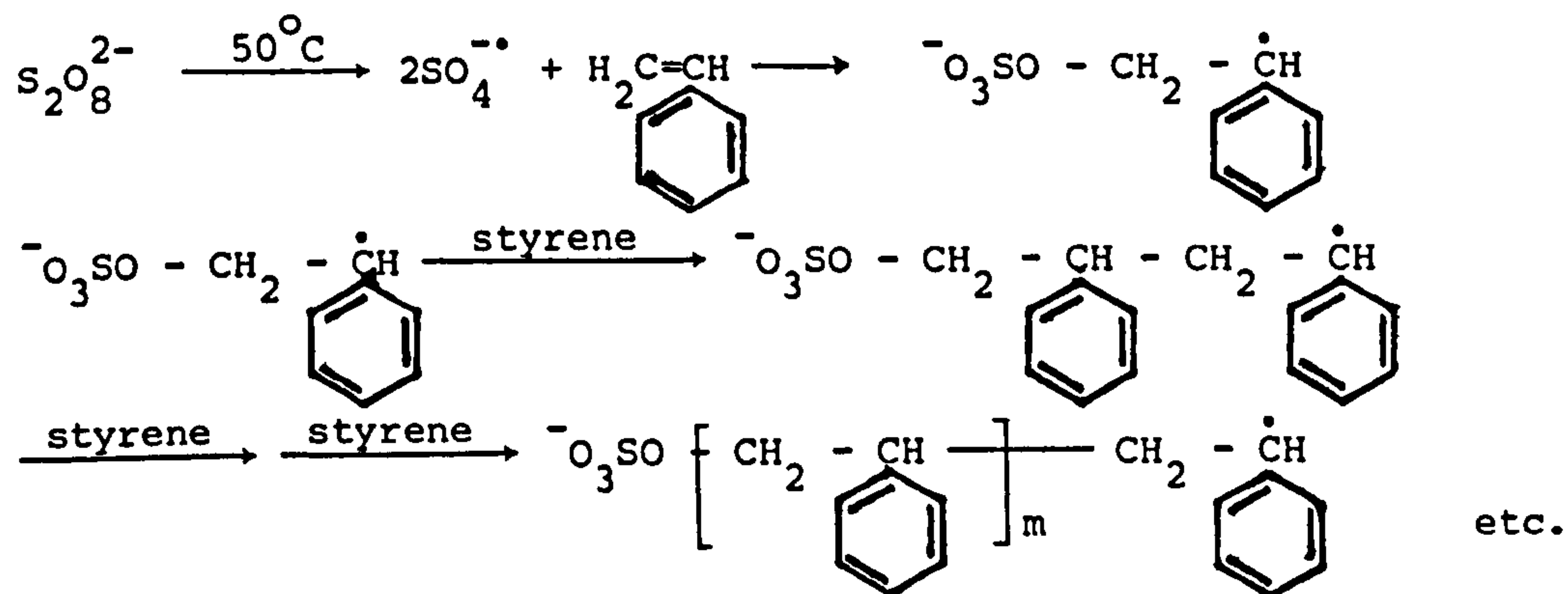
The model system chosen for investigation in this work was cross-linked polystyrene swollen microgel particles dispersed in non-polar media, such as toluene or 1,4-dioxane. The preparation of the particles is essentially the same as for "normal" polystyrene latex particles (i.e. surfactant-free emulsion polymerisation in water). The addition of cross-linking agent prevents the particles from dissolving when redispersed into, for example, toluene. Instead, the particles swell. Such systems may be considered thermodynamically stable<sup>121</sup>, and, as such, are ideal for studying surface charge effects in non-polar media, since no steric stabilisers need to be employed.



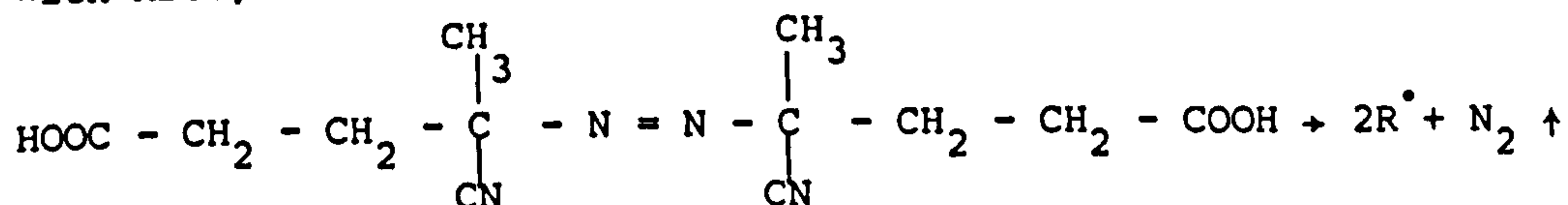
In order to prepare microgel particles, it is necessary to perform the polymerisation in a poor solvent for polystyrene, to promote nucleation, and a highly polar solvent, to promote electrostatic stabilisation. Hence, the route chosen for the preparation of the microgel systems in this work is the aqueous route described by Clarke<sup>122</sup>.

The ionogenic surface groups resulting from the polymerisation depend on the initiator used. In this work, two initiators were used; potassium persulphate,  $K_2S_2O_8$ , resulting in surface sulphate groups, and 4,4'-azo-bis (cyanovaleric acid), ABCV, resulting in surface carboxylic acid groups.


The general reaction scheme with  $K_2S_2O_8$  as initiator is<sup>123</sup>



With ABCV,



then free radical polymerisation occurs as with  $\text{SO}_4^{\cdot-}$  to yield polystyrene chains with terminal  $-\text{COOH}$  groups. Divinyl benzene

( $\text{H}_2\text{C} = \text{CH}$    $\text{CH} = \text{CH}_2$ , DVB) is used as the cross-linking agent.

According to Guthrie<sup>124</sup>, when the growing polymer chains reach a critical length, self-nucleation occurs, generating primary particles. These primary particles aggregate until the surface charge density prevents further aggregation. Thus, by performing the polymerisation in an electrolyte solution, larger particles can be produced, due to the electrostatic screening of the particles by the electrolyte.

Two microgel samples are described in this work, one possessing surface sulphate groups (designated PSLX-PP-1) and the other possessing surface carboxylic groups (designated PSLX-A-2).

#### 5.2.2 PREPARATION OF PSLX-PP-1

550ml of water and 0.7g of sodium chloride were stirred in a 1 litre, 3-neck, round-bottomed flask in an oil bath at  $70^\circ\text{C}$  for 15 minutes. This solution was degassed by gently blowing dry nitrogen through the reaction vessel. 75.0ml of styrene and 0.5ml of DVB were then added, and the contents of the flask stirred for a further 15 minutes. 0.243g of  $\text{K}_2\text{S}_2\text{O}_8$ , dissolved in 30ml of water, was then added to the contents of the flask. The reaction was allowed

to proceed for 24 hours. After this time, the resultant white latex was first filtered through glass wool, to remove any large aggregates, and then cleaned by repeated centrifugation/redispersal cycles into water. Finally, the latex was dialysed against water for two weeks.

### 5.2.3 PREPARATION OF PSLX-A-2

The same general procedure was used here as for PSLX-PP-1, except that 600cm<sup>3</sup> of H<sub>2</sub>O were used at pH 9.0 (using sodium hydroxide to adjust the pH). 23.5cm<sup>3</sup> of styrene and 0.5cm<sup>3</sup> DVB were added to the water in the flask, and the contents stirred for 3/4 hour. 0.0660g of ABCV dissolved in 30cm<sup>3</sup> of water at pH 11 were added to initiate the reaction.

### 5.2.4 REDISPERSAL OF MICROGELS INTO NON-POLAR MEDIA

The microgel particles were redispersed into non-polar media by repeated centrifugation/redispersal cycles. Freeze-drying of the particles from aqueous suspension and redispersal into non-polar media was found to result in small aggregates of particles by optical microscopy. The former method, however, yielded dispersions of discrete particles.

### 5.3 ELECTROPHORETIC STUDIES OF PSLX-PP-1 IN TOLUENE

The electrophoretic mobility for PSLX-PP-1 particles dispersed in toluene was measured by PALS as a function of the water concentration in the dispersion medium. The particle diameter and initial and equilibrium water concentrations were determined for each sample.

#### 5.3.1 GRAVIMETRIC ANALYSIS OF "PARENT" PSLX-PP-1/TOLUENE DISPERSION

Table 5.1 shows the data for the gravimetric analysis of the "parent" dispersion used for the study.

From the data in table 5.1, the mass fraction of solids is calculated to be  $0.95 \pm 0.11\%$ .

#### 5.3.2 SAMPLE PREPARATION AND MEASUREMENT

Three batches of toluene, each with different water contents (10 ppm, 90 ppm and 180 ppm by mass), were used to prepare 20ml quantities of toluene with water concentrations ranging from 17.6 ppm to 165.4 ppm. 1.0ml of each sample was used for water content determination. Table 5.2 shows the water contents for each of the dispersion media. These figures are taken to be the initial water contents for each of the samples. To each of the toluene samples



Mass of vessel/g	Mass of vessel and sample/g	Total mass after drying/g	Mass fraction of solids
1.4910	2.4434	1.4992	0.0086
1.4112	2.0447	1.4173	0.0096
1.4696	2.2697	1.4764	0.0085
1.3753	2.1971	1.3833	0.0097
1.4074	2.2448	1.4167	0.0111

Table 5.1: Data for gravimetric analysis of PSLX-PP-1/  
toluene dispersions.

Sample	Water content /ppm	Sample	Water content /ppm
1	17.6	6	94.2
2	33.1	7	110.3
3	47.3	8	127.8
4	62.8	9	152.3
5	77.4	10	165.4
		11	180.0

Table 5.2: Initial dispersion medium water contents,  
by mass, for PSLX-PP-1 in toluene.

were added 0.2ml of the concentrated dispersion. The electrophoretic mobility for each sample was determined within 5 minutes of adding the particles to the toluene. The mobilities were determined using two field strengths, 50V across 2.75 mm and 100V across 2.75 mm. The particle diameters were determined using PCS. The samples were allowed to equilibrate for 18 hours, after which time their diameters, mobilities and water contents were determined.

Table 5.3 shows the initial mobilities and particle diameters for each of the samples. Table 5.4 shows the equilibrium mobilities, diameters and water contents for each of the samples. It is evident from the PCS data that the particle diameter is not influenced by the water concentration. An average diameter of  $1163 \pm 39$  nm is obtained for PSLX-PP-1 particles dispersed in toluene.

Figures 5.2(a) and 5.2(b) show the initial and equilibrium mobilities for the samples as a function of initial water concentration, respectively. Figure 5.3 shows how the equilibrium water content varies as a function of the initial water content. For each sample, approximately 50% of the water initially present in the bulk toluene is taken up by the particles. The mobility data show that there is little change in mobility after 18 hours for the wetter (>75ppm) samples, but there is a noticeable change for the drier samples, especially at ~ 50 ppm initial water content. This may suggest that the kinetics of the partitioning of the water is rapid (within a few minutes) for the wetter samples, but relatively slow for the much drier samples.



Sample	Mobility		Diameter /nm
	$/10^{-11} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$		
	$E_{\text{O}} = 50\text{V}/2.75\text{mm}$	$E_{\text{O}} = 100\text{V}/2.75\text{mm}$	
1	$-15.7 \pm 2.5$	$-13.1 \pm 0.8$	$1128 \pm 43$
2	$-5.36 \pm 1.44$	$-4.92 \pm 1.13$	$1149 \pm 33$
3	$+0.52 \pm 2.70$	$+0.55 \pm 1.09$	$1169 \pm 12$
4	$-9.70 \pm 1.32$	$-10.1 \pm 1.02$	$1167 \pm 56$
5	$-13.3 \pm 0.9$	$-12.9 \pm 0.9$	$1144 \pm 30$
6	$-14.4 \pm 3.0$	$-11.8 \pm 0.7$	$1170 \pm 40$
7	$-10.9 \pm 1.42$	$-10.7 \pm 1.21$	$1137 \pm 21$
8	$-12.0 \pm 1.31$	$-11.1 \pm 1.26$	$1127 \pm 33$
9	$-10.1 \pm 1.30$	$-13.0 \pm 1.15$	$1141 \pm 26$
10	$-10.9 \pm 1.26$	$-10.8 \pm 0.75$	$1130 \pm 17$
11			-

Table 5.3: Initial mobilities and diameters for PSLX-PP-1 in toluene as a function of dispersion medium water content. (See Table 5.2 for water contents).

Sample	Mobility		Water Conc <sup>n</sup>	Diameter
	$/10^{-11} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$		$/\text{ppm}$	$/\text{nm}$
	$E_{\text{O}}=50\text{V}/2.75\text{mm}$	$E_{\text{O}}=100\text{V}/2.75\text{mm}$		
1	$-7.55 \pm 1.29$	$-11.2 \pm 0.16$	$25.8 \pm 3.9$	$1169 \pm 34$
2	-	$-6.79 \pm 0.55$	$25.6 \pm 1.5$	$1167 \pm 51$
3	$-10.8 \pm 2.0$	$-9.67 \pm 1.21$	$29.5 \pm 4.4$	$1144 \pm 31$
4	$-12.6 \pm 1.11$	$-13.1 \pm 1.46$	$35.5 \pm 0.1$	$1185 \pm 61$
5	$-14.9 \pm 1.81$	$-13.4 \pm 0.67$	$48.9 \pm 3.3$	$1216 \pm 45$
6	$-9.66 \pm 3.01$	$-10.4 \pm 0.77$	$55.3 \pm 2.1$	$(1235 \pm 44)$
7	$-10.7 \pm 1.58$	$-10.6 \pm 1.70$	$60.6 \pm 0.6$	$1196 \pm 27$
8	$-13.0 \pm 0.83$	$-12.9 \pm 3.81$	$68.7 \pm 6.4$	$1162 \pm 56$
9	$-12.4 \pm 1.52$	$-14.2 \pm 1.77$	$72.5 \pm 2.4$	$1191 \pm 32$
10	$-10.7 \pm 2.07$	$-12.1 \pm 0.40$	$83.2 \pm 0.3$	$1186 \pm 29$
11			-	-

Table 5.4: Mobilities, diameters and water contents (in bulk phase) for equilibrated PSLX-PP-1/toluene samples.

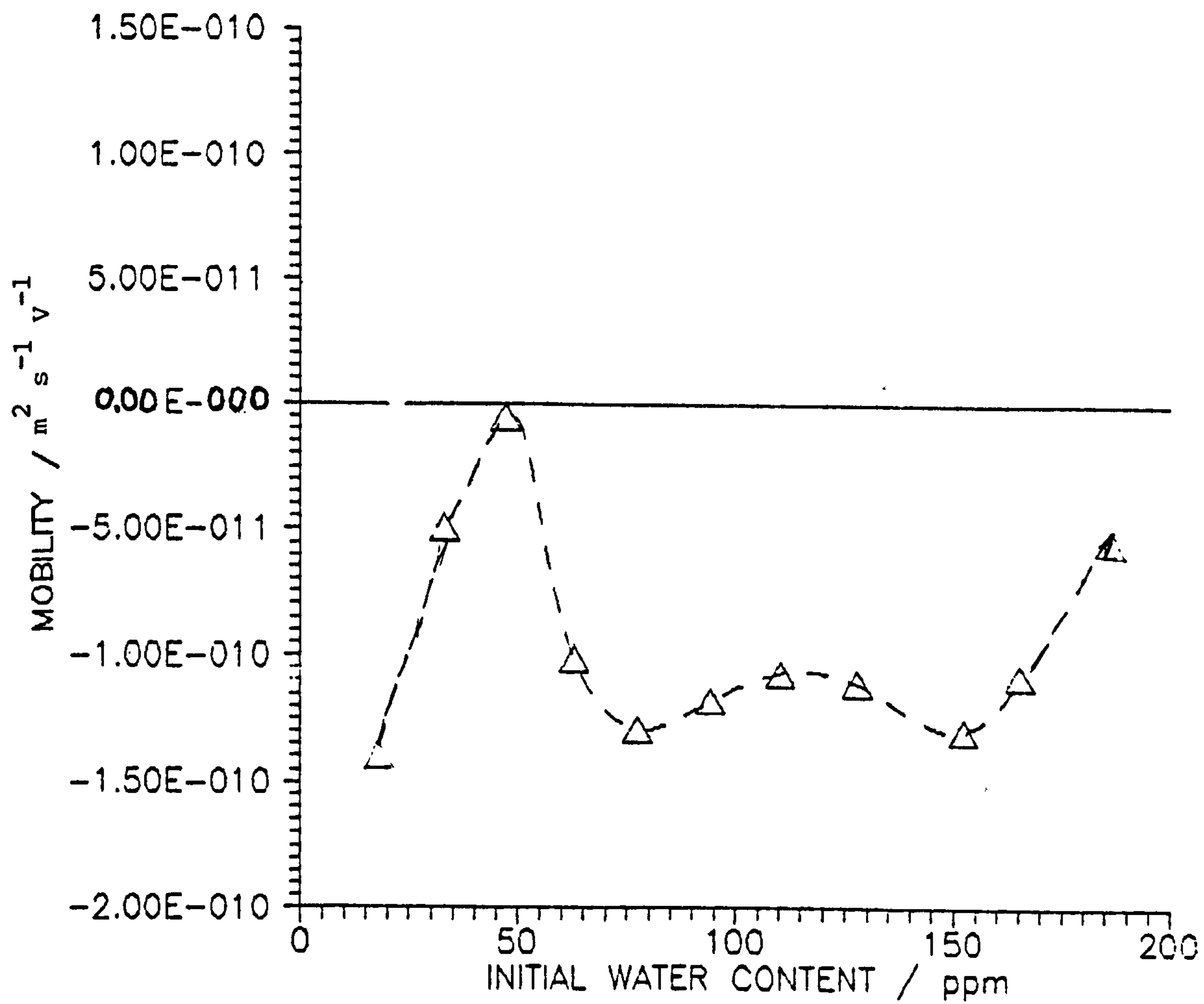


Figure 5.2 (a): Initial electrophoretic mobility for PSLX-PP-1 dispersed in toluene as a function of trace water content of dispersion medium.

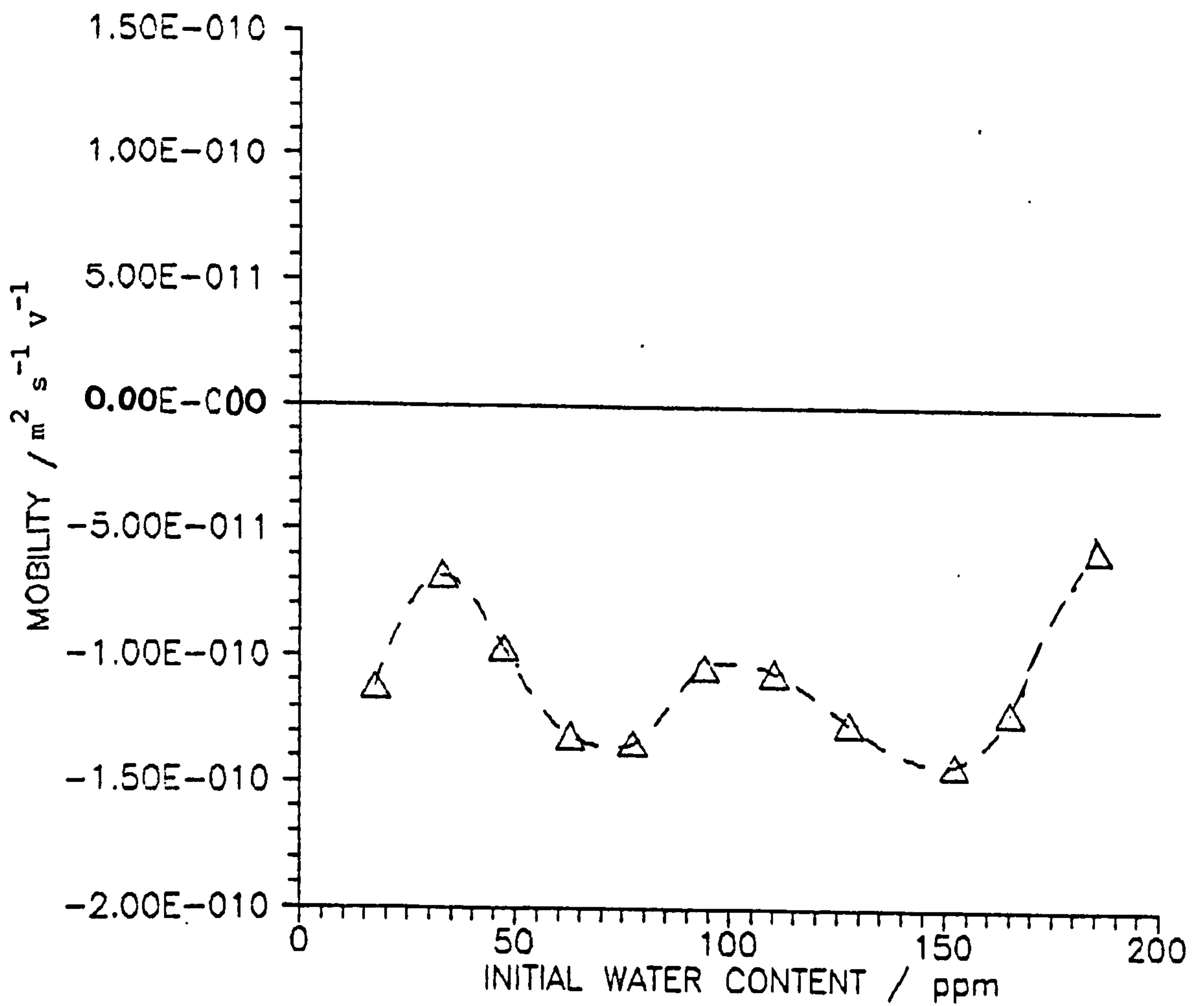


Figure 5.2 (b): Equilibrium electrophoretic mobility for PSLX-PP-1 dispersed in toluene as a function of trace water content of dispersion medium.

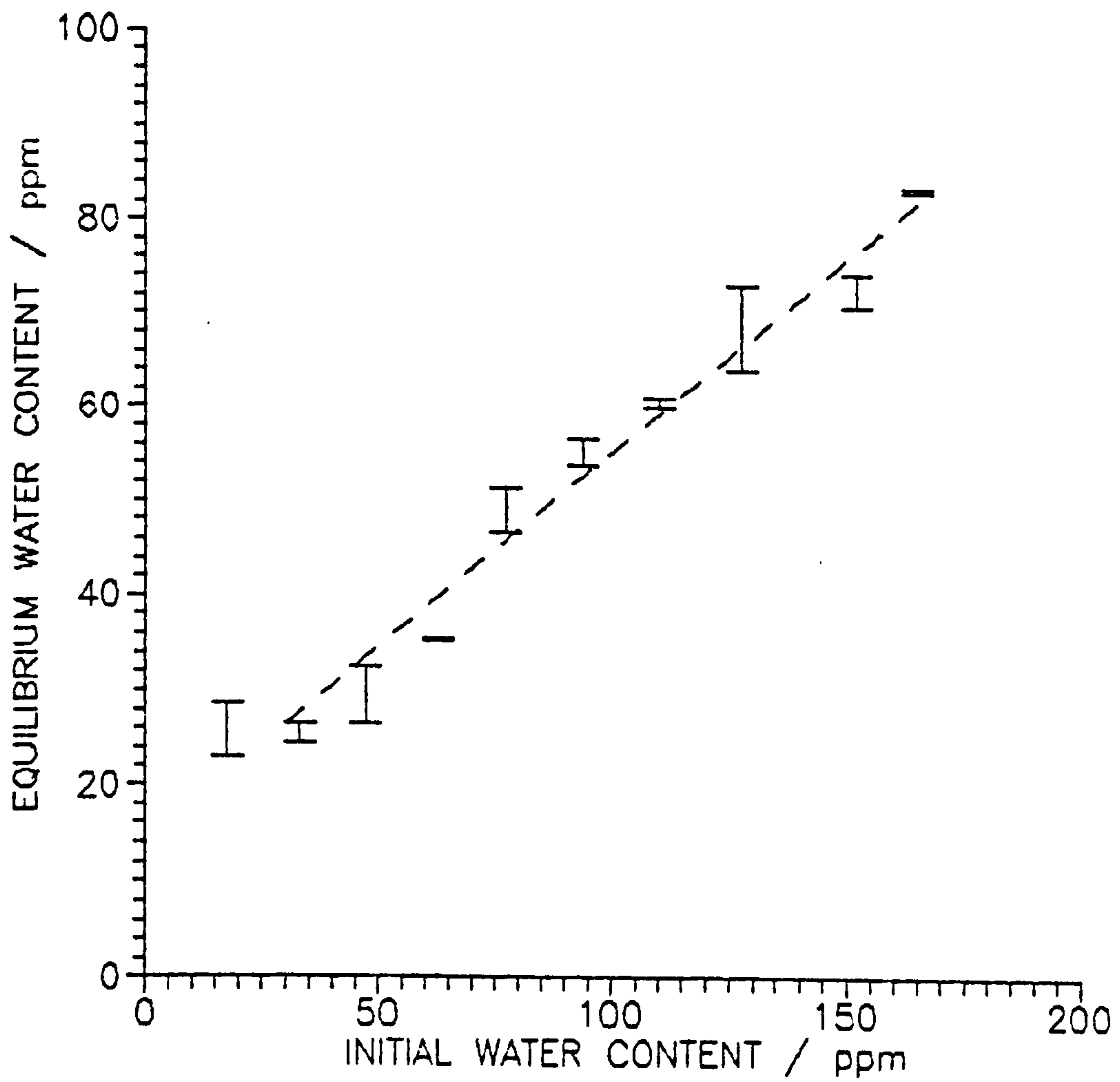


Figure 5.3:

Equilibrium bulk water concentration vs. initial bulk water concentration for PSLX-PP-1 particles dispersed in toluene.

In order to investigate this further, 0.2ml of the concentrated latex was added to 19ml of toluene containing  $18 \pm 4$  ppm water, and the mobility of the resulting dispersion was measured as a function of time. The results are given in table 5.5(a). The mobility shows a slight increase over a period of 4 hours. The experiment was repeated, yielding the data shown in table 5.5(b). This time, the observed behaviour is entirely different, with the mobility exhibited after 3 hours being 3 times higher than that for the previous experiment. The only difference between the two experiments was that, for the latter case, the sample bottle was regularly agitated vigorously, allowing the sample to come into contact with the septum used to seal the sample from the atmosphere.

Clearly, the contact of the sample with the septum has an adverse effect on the electrophoretic mobility of the sample. In order to see if the extent of the effect depends on the dryness of the dispersion medium, the experiment was repeated using toluene containing 180 ppm water by mass. Table 5.6(a) shows the mobility obtained as a function of time for the sample with no contact with the septum, and table 5.6(b) shows the mobility for the sample being shaken for 15 seconds between each mobility measurement. Thus, the effect, if any, for the wetter sample is less pronounced than for the drier one.

The samples used to obtain the data in tables 5.3 and 5.4 were used to see how contact of the samples with the septa affected the



Time after addition of latex/min	Electrophoretic mobility/ $10^{-11} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$
3.0	$-14.7 \pm 0.86$
40.5	$-17.6 \pm 1.11$
70.5	$-11.8 \pm 1.25$
96.5	$-15.7 \pm 1.66$
144.0	$-15.0 \pm 1.18$
189.5	$-16.3 \pm 1.10$
208.5	$-16.6 \pm 1.62$
224.5	$-17.9 \pm 0.39$
234.5	$-18.8 \pm 0.84$

Table 5.5(a): Electrophoretic mobility of PSLX-PP-1 in toluene  
( $18 \pm 4$  ppm water) as a function of time after addition  
of particles to toluene.

Time after addition of latex/min	Electrophoretic mobility/ $10^{-11} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$
8	$-16.5 \pm 0.45$
47	$-36.4 \pm 1.70$
78	$-44.0 \pm 3.62$
114	$-45.4 \pm 1.57$
148	$-48.4 \pm 1.53$
180	$-43.3 \pm 1.28$

Table 5.5(b): Repeated determination of mobility of PSLX-PP-1 in toluene  
( $18 \pm 4$  ppm water) as a function of time after addition  
of particles to toluene.

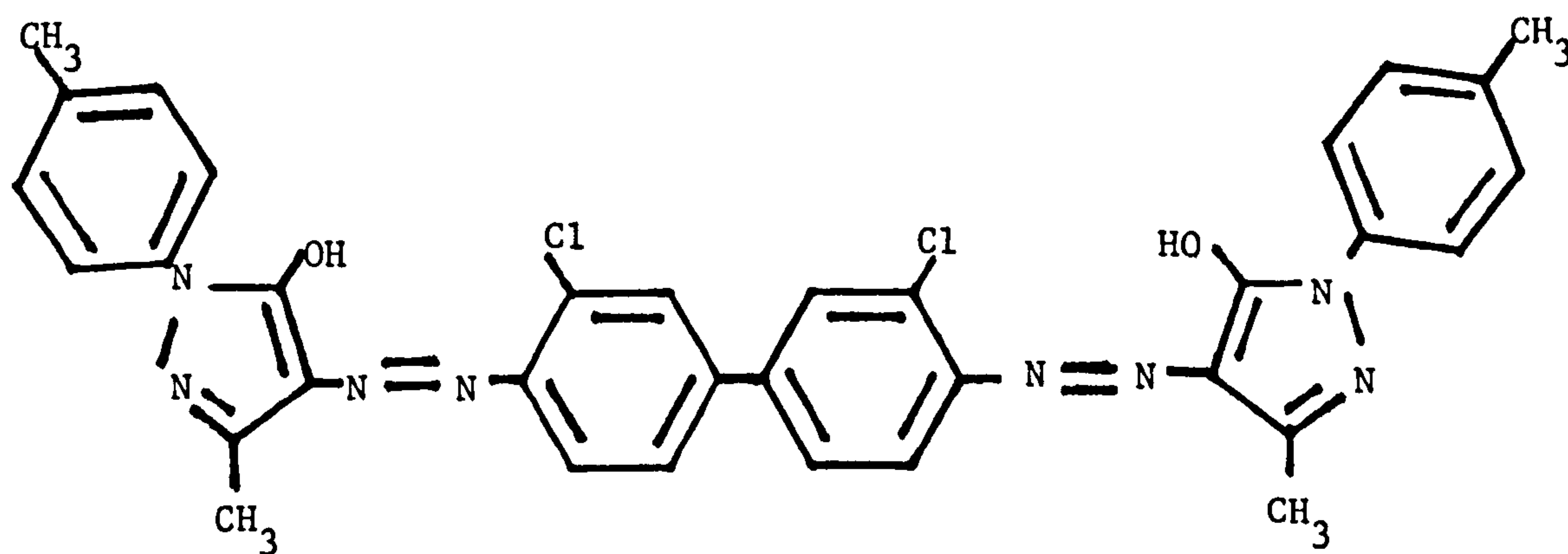
Time after addition of latex/min	electrophoretic mobility/ $10^{-11} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$
6	$-5.15 \pm 1.70$
20	$-4.18 \pm 0.90$
55	$-6.21 \pm 0.59$
79	$-5.16 \pm 1.00$
116	$-5.23 \pm 1.21$

Table 5.6(a): Mobility of PSLX-PP-1 in toluene (180 ppm water)  
as a function of time, with no septum contact.

Time after addition of latex/min	Electrophoretic mobility/ $10^{-11} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$
6	$-3.61 \pm 1.25$
28	$-4.19 \pm 0.91$
57	$-5.10 \pm 1.06$
92	$-6.50 \pm 1.45$
134	$-6.47 \pm 0.93$

Table 5.6(b): Mobility of PSLX-PP-1 in toluene (180 ppm water)  
as a function of time, with septum contact.

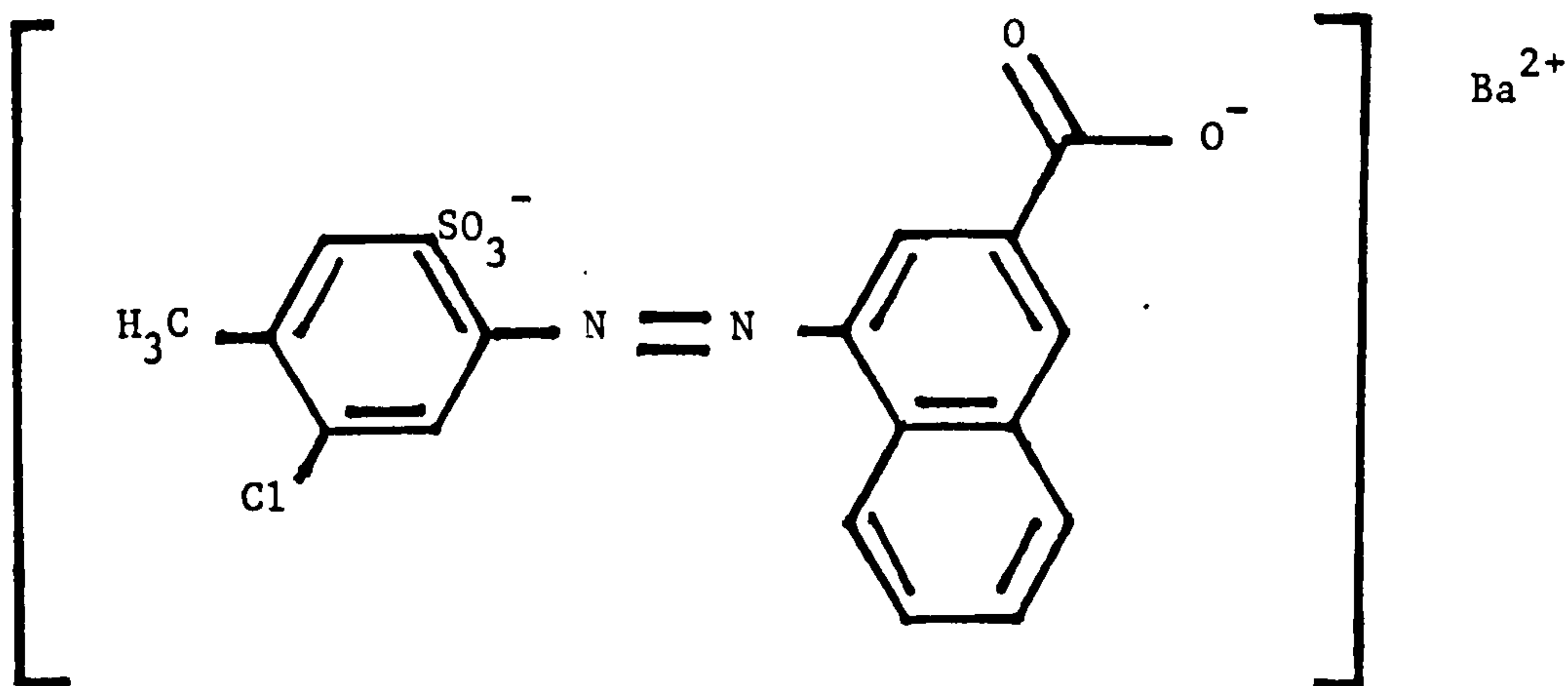
electrophoretic mobility. Table 5.7 shows the mobility of the samples after one week from preparation. In each case, the mobility was measured before shaking for 90 seconds and then afterwards to see how the mobility changed. It should be mentioned that, by this stage, the repeated insertion and removal of syringe needles through the septa had left small, but visible "cores" from the septa in the dispersions. Hence, the initial mobilities are not the same as given in Table 5.4. The effects of contact with the septa are clearly unpredictable, with some samples showing a decrease in mobility and others an increase in mobility. The cause of the change in mobility is likely to arise from the extraction of plasticisers and/or pigments from the rubber septa. It was noticed that the samples acquired an orange tinge after shaking. The red septa are known to contain the following two pigments<sup>125</sup>.



"irgalite orange F2G" (CI Pigment orange 21115)

Sample	Mobility before $/10^{-11} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$	Mobility after $/10^{-11} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$	% increase
1	$-6.08 \pm 0.74$	$-10.5 \pm 0.72$	73
2	$-2.38 \pm 0.46$	$-4.01 \pm 0.79$	68
3	$-7.55 \pm 0.57$	$-11.4 \pm 0.28$	51
4	$-7.47 \pm 0.83$	$-10.3 \pm 0.43$	38
5	$-7.72 \pm 0.35$	$-20.8 \pm 1.09$	169
6	$-9.86 \pm 0.49$	$-8.90 \pm 0.30$	-10
7	$-8.66 \pm 0.12$	$-10.0 \pm 0.88$	15
8	$-8.64 \pm 0.64$	$-7.07 \pm 1.08$	-18
9	$-5.88 \pm 0.47$	$-15.5 \pm 1.38$	164
10	$-10.8 \pm 0.56$	$-16.8 \pm 0.48$	56

Table 5.7: Mobility data for 1 week old PSLX-PP-1/toluene samples before and after shaking in contact with septum.



(CI Pigment Red 15865:1).

The presence of either of these two compounds in the PSLX-PP-1/toluene samples could have a strong influence on the interfacial charge/potential developed.

In view of the undesirable effects of contamination of the samples by materials in the rubber septa, it was decided that the study of the effects of trace amounts of water in the systems described here could not be investigated any further until an alternative method of preparing and storing the samples became available.

#### 5.4 STUDY OF PSLX-PP-1 AND PSLX-A-2 IN DIOXAN/WATER MIXTURES

As a result of the problems discussed above, it was decided to investigate the electrophoretic mobility and particle size variation for PSLX-PP-1 and PSLX-A-2 dispersed (separately) in water/dioxan mixtures.



11 water/dioxan mixtures were prepared, ranging from 100% dioxan to 100% water. Each of these were divided into two portions and sufficient quantity of latex added (from a dispersion in pure dioxan) to yield a series of dispersions with ~ 0.01% mass fraction of particles for PSLX-PP-1 and PSLX-A-2.

Table 5.8 shows the physical parameters for the solvent mixtures used in this study. The refractive indices were obtained using the Abbé refractometer, whereas the relative permittivity and viscosity data were obtained from the literature<sup>118</sup>.

For each series of samples, the mobility was determined within 5 minutes of preparation and again 24 hours later. The particle diameters were obtained for the 24 hour old samples using PCS. In the following results, the PSLX-PP-1 samples are referred to as WDPPn, where n is the mixture number, given in table 5.8. Similarly, WDAn is used to identify PSLX-A-2 samples.

Table 5.9 shows the data obtained for both series of samples. Figure 5.4 is a graphical representation of the equilibrium mobilities for both series. The variation in particle size is shown graphically in figure 5.5. The calculated equilibrium zeta potentials (using eq. 2.45), equivalent electrophoretic charges per particle and surface charge densities are shown for both series graphically in figures 5.6, 5.7 and 5.8, respectively.

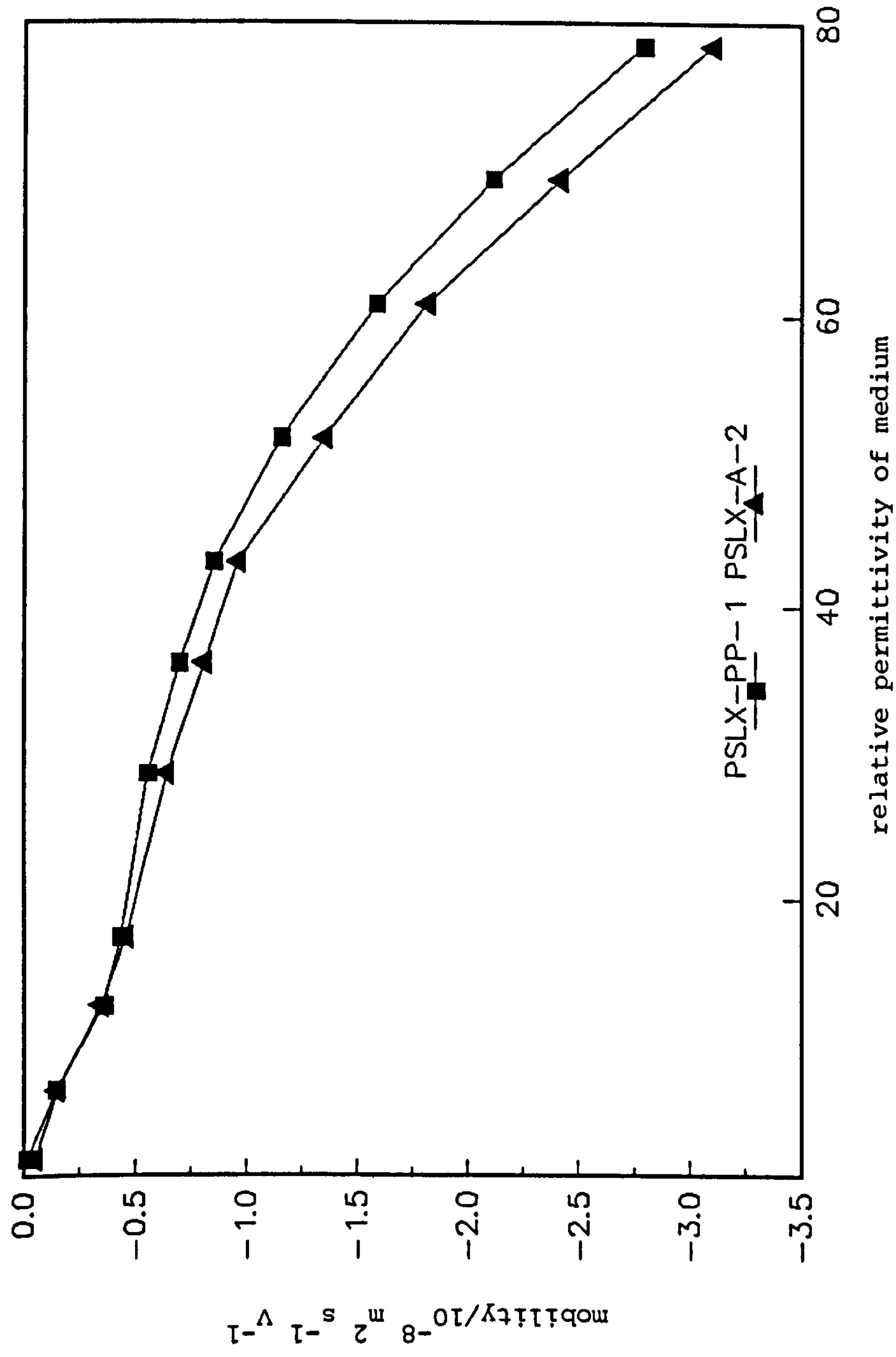


Mixture No.	% water By mass	Viscosity /10 <sup>-3</sup> kg m <sup>-1</sup> s <sup>-1</sup>	Relative Permittivity	Ref. Index
1	0.00	1.09	2.2	1.422
2	9.85	1.40	6.9	1.413
3	21.00	1.75	12.7	1.403
4	27.61	1.89	17.4	1.397
5	41.53	1.98	28.6	1.385
6	50.23	1.91	36.2	1.377
7	58.35	1.75	43.1	1.369
8	68.89	1.54	51.6	1.360
9	78.88	1.32	60.8	1.351
10	89.01	1.11	69.3	1.343
11	100.00	0.89	78.5	1.332

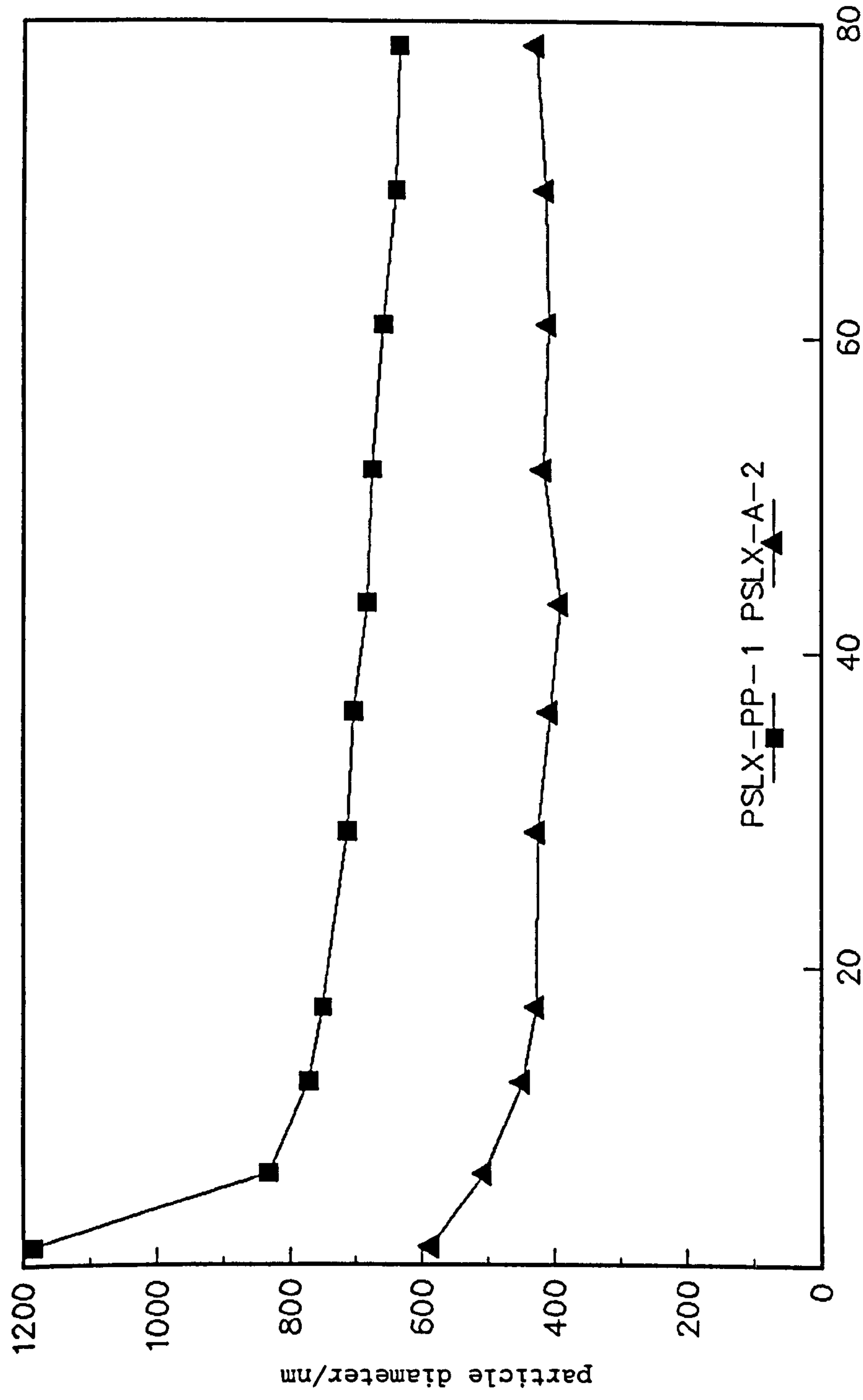
Table 5.8: Physical parameters for water/dioxan mixtures  
at 25<sup>o</sup>C.

Sample	Mobility/ $10^{-10} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$		Zeta potential/mV		Particle
Code	Fresh	Equilib.	Fresh	Equilib.	Diam./nm
WDPP1	-1.71±0.17	-1.86±0.12	-14.3±1.4	-15.6±1.0	1184±41
WDPP2	-15.3±0.2	-14.8±0.5	-52.6±0.7	-50.9±1.7	831±26
WDPP3	-36.9±0.8	-36.2±0.5	-86.1±1.9	-84.5±1.2	771±36
WDPP4	-44.0±1.3	-43.5±0.5	-81.0±2.4	-80.1±0.9	749±7
WDPP5	-60.5±0.5	-56.1±0.4	-70.9±0.5	-65.8±0.5	712±33
WDPP6	-70.2±0.5	-70.2±1.4	-62.7±0.4	-62.8±1.3	703±21
WDPP7	-90 ±3	-85.7±2.7	-61.9±2.1	-59.0±1.9	682±14
WDPP8	-122 ±3	-116 ±2	-61.7±1.5	-58.7±1.0	675±7
WDPP9	-161 ±3	-159 ±3	-59.2±1.1	-58.5±1.1	658±15
WDPP10	-227 ±5	-212 ±2	-61.6±1.4	-57.5±0.5	638±18
WDPP11	-308 ±3	-280 ±1	-59.2±0.6	-53.8±0.2	633±14
.					
WDA1	-6.39±0.10	-5.21±0.17	-53.5±0.8	-43.6±1.4	587±16
WDA2	-13.5±0.2	-15.2±0.3	-46.4±0.7	-52.2±1.0	506±9
WDA3	-34.0±0.3	-34.8±0.9	-79.4±0.7	-81.2±2.1	448±8
WDA4	-46.9±1.4	-45.8±0.3	-86.3±2.6	-84.3±0.6	427±13
WDA5	-70.5±0.9	-64.7±0.5	-82.7±1.1	-75.9±0.6	426±8
WDA6	-84.4±1.4	-81.3±1.0	-75.4±1.3	-72.7±0.9	407±4
WDA7	-98.8±0.9	-96.6±0.8	-68.0±0.6	-66.4±0.6	391±14
WDA8	-134 ±3	-136 ±1	-67.8±1.8	-68.8±0.5	416±5
WDA9	-185 ±5	-182 ±5	-68.0±1.8	-66.9±1.8	408±15
WDA10	-247 ±3	-242 ±2	-67.0±0.8	-65.7±0.5	412±10
WDA11	-302 ±4	-311 ±5	-58.1±0.8	-59.7±1.0	426±6

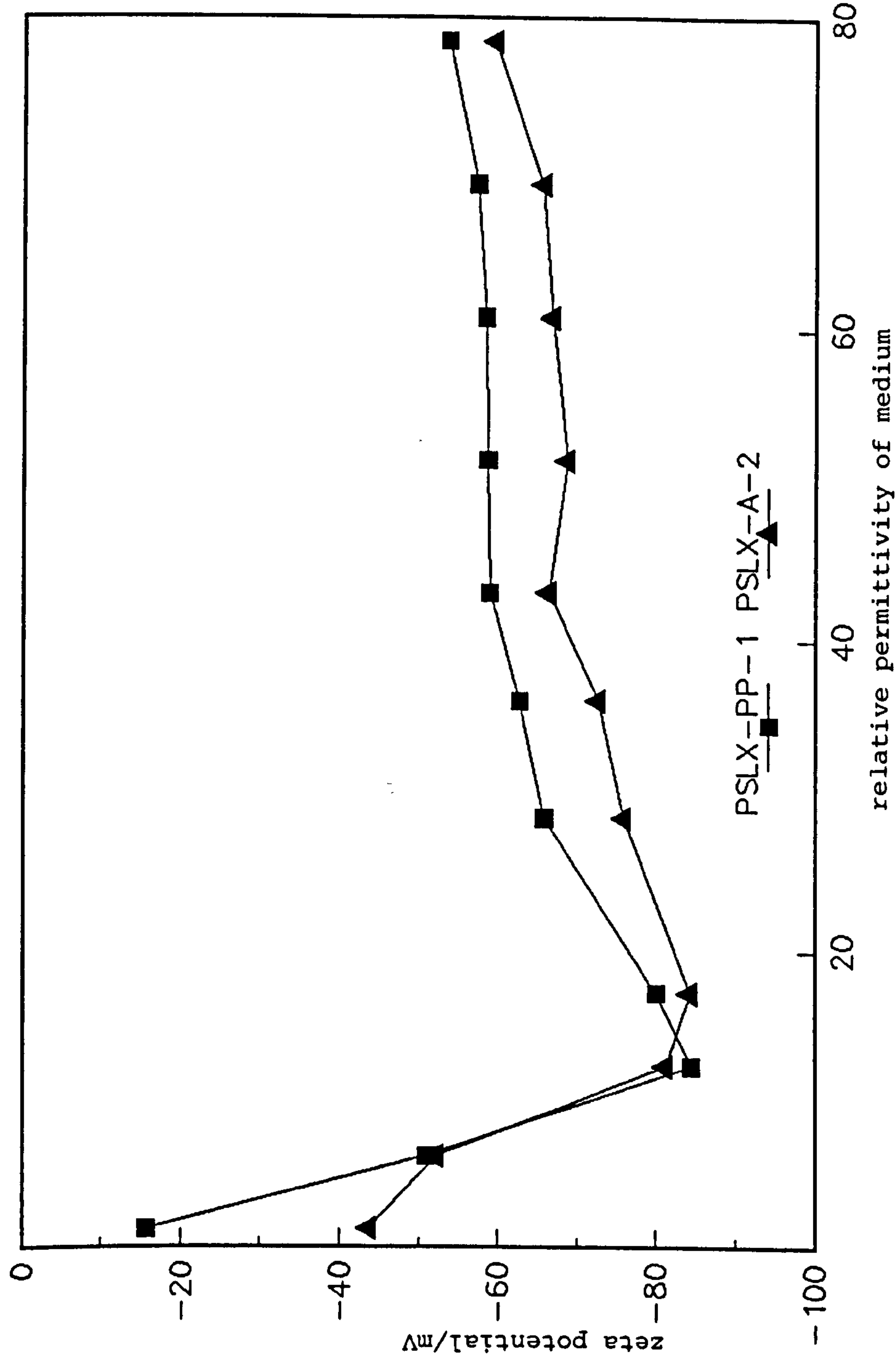
Table 5.9:      Mobility, zeta potential and diameter results for  
PSLX-PP-1 and PSLX-A-2 in water/dioxan mixtures.  
(See text for further details).



**Figure 5.4:** Equilibrium electrophoretic mobilities for two cross-linked polystyrene microgel samples dispersed in water/dioxan as a function of dispersion medium relative permittivity.



**Figure 5.5:** Particle diameters (determined using PCS) for cross-linked polystyrene microgel samples dispersed in water/dioxan as a function of dispersion medium relative permittivity.



**Figure 5.6:** Calculated equilibrium zeta potential (using Hückel equation) for two cross-linked polystyrene microgel samples dispersed in water/dioxan as a function of dispersion medium relative permittivity.

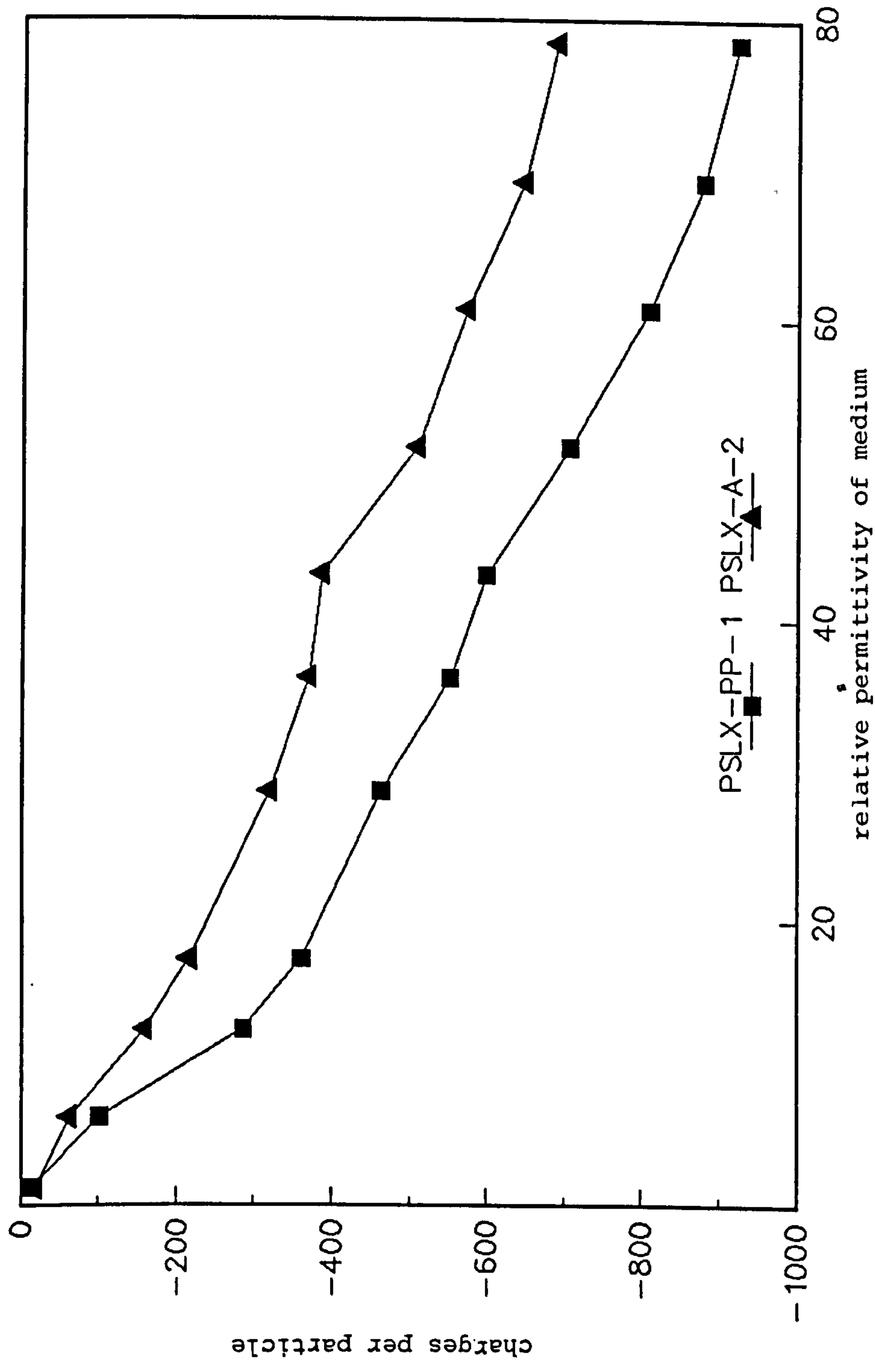


Figure 5.7: Calculated equivalent electrophoretic charges per particle for two cross-linked polystyrene microgel samples dispersed in water/dioxan as a function of dispersion medium relative permittivity.



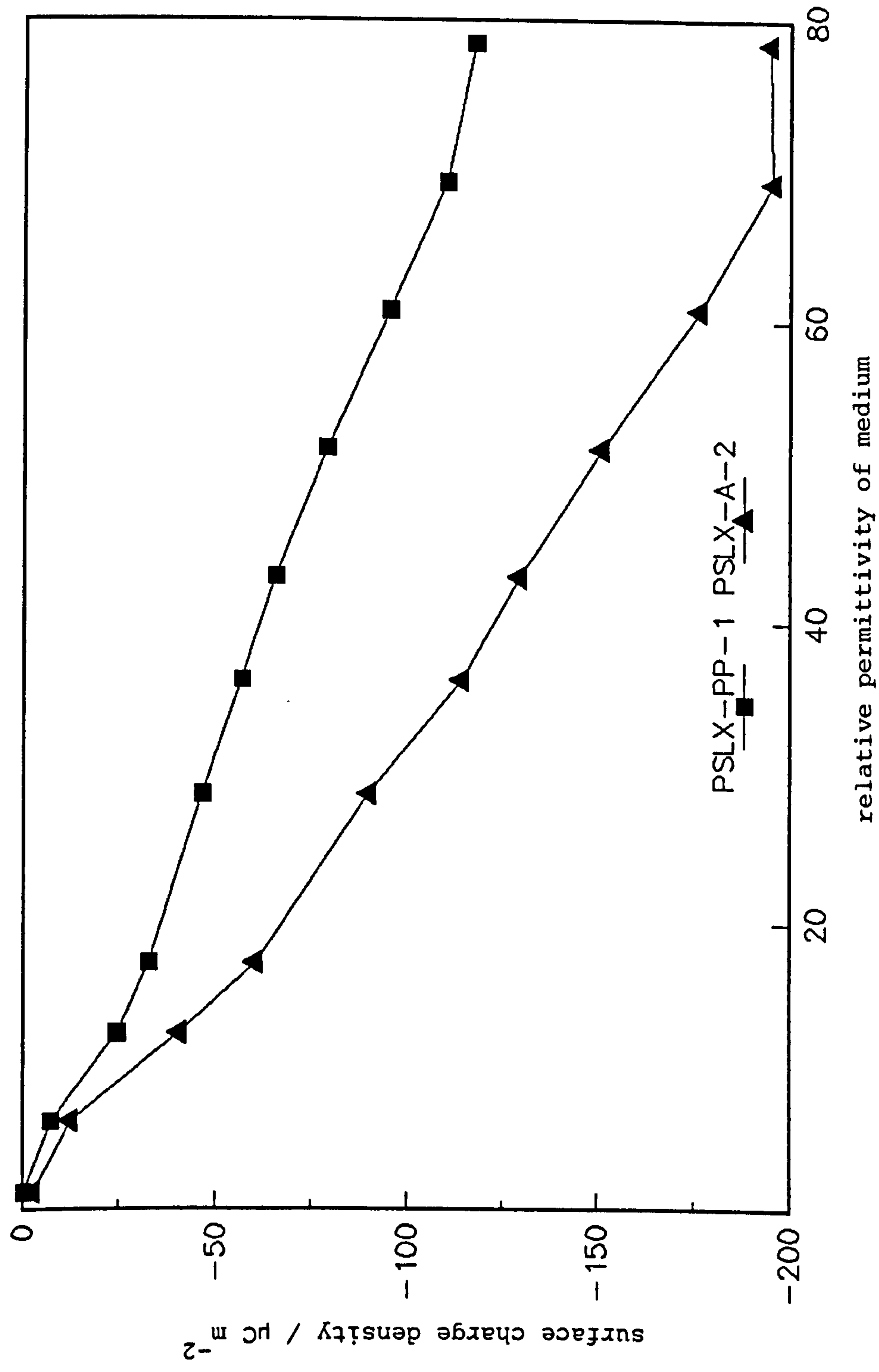


Figure 5.8: Calculated electrophoretic surface charge densities for two cross-linked polystyrene microgel samples dispersed in water/dioxan as a function of dispersion medium relative permittivity.

The results obtained show some interesting features. Firstly, although the two types of particle have different surface groups and diameters, their mobilities across the range of media are fairly similar, showing the same general trend. Only the samples in 100% dioxan have appreciably different mobilities. The manner in which the particles swell as the media become better solvents for polystyrene is noteworthy for both systems. The particle size increases slowly as the medium composition changes from 100% water to 20 - 25% water (by mass). As the medium becomes more rich in dioxan, the particles swell to a much greater extent (figure 5.5). The PSLX-PP-1 particles swell much more than the PSLX-A-2 particles because the former particles were prepared using one third of the quantity of cross-linking agent used for the PSLX-A-2 preparation. The point at which the particles begin to swell more easily appears to coincide with the maxima observed for the zeta potentials (figure 5.6). The opening of the microgel particles is accompanied by a rapid decrease in the calculated zeta potentials. However, this decrease in zeta potential is less for the PSLX-A-2 particles, which do not swell as much as the PSLX-PP-1 particles

The zeta potentials appear to remain constant over a fairly broad range of medium composition (~60 - 90% water by mass), although the calculated equivalent electrophoretic surface charge densities show an almost linear decrease across the whole range of media (Figure 5.8).

It is interesting to consider what might happen if the degree of cross-linking in the microgel particles was sufficient to maintain a constant particle size across the whole range of medium compositions. Would a maximum in the zeta potential be observed, or would the potential continue to rise, to reach a limiting value?

#### 5.5 STUDY OF PSLX-PP-1 IN DIOXAN/WATER/KCl SOLUTIONS

The electrophoretic mobility of PSLX-PP-1 particles dispersed in dioxan/water mixtures was determined as a function of the concentration of potassium chloride in solution. The presence of electrolyte allows zeta potentials to be calculated using the O'Brien and White method (§ 2.4.2) and the Ohshima method (eqs. 2.54 to 2.60), as well as the simpler Hückel and Smoluchowski relationships.

Table 5.10 shows the physical parameters for the water/dioxan/KCl solutions used<sup>126</sup>. The conductivity of the electrolyte at infinite dilution,  $\Lambda_0$ , and the electrolyte association constant,  $K_A$ , are required for the determination of  $\zeta$  when using the O'Brien and White or Ohshima methods. For each of the 9 water/dioxan media (A to I), a stock solution of  $10^{-1}$  mol dm<sup>-3</sup> potassium chloride was prepared. Serial dilution of these solutions generated  $10^{-2}$ ,  $10^{-3}$ ,  $10^{-4}$  and  $10^{-5}$  mol dm<sup>-3</sup> KCl solutions in water/dioxan. For the following

Medium	% water by mass	$\epsilon_r$	$\Lambda_O$ /S cm <sup>-1</sup>	$K_A$	$\eta$ /10 <sup>-3</sup> kg m <sup>-1</sup> s <sup>-1</sup>	d /nm
A	20.0	12	38	1700	1.71	775
B	30.5	17	44	251	1.93	750
C	40.5	25	52	32	1.98	717
D	49.5	36	61	6.3	1.91	704
E	59.8	43	72	2.5	1.72	680
F	69.8	52	89	1.8	1.50	665
G	79.4	62	108	1	1.28	658
H	89.6	69	123	1	1.11	646
I	100.0	78	149	1	0.89	634

Table 5.10: Physical parameters for water/dioxan/potassium chloride solutions at 25<sup>o</sup>C.  $\epsilon_r$ ,  $\Lambda_O$ ,  $K_A$ ,  $\eta$  and d are the relative permittivity, conductivity at infinite dilution, association constant, viscosity and particle diameter (i.e. of PSLX-PP-1), respectively.



results, each medium is referred to by an alphanumeric code consisting of a letter prefix, A to I, corresponding to the code in table 5.10, and a numeric suffix, 1 to 5, representing the exponent of the concentration of KCl in solution (i.e. B3 refers to a solution of  $10^{-3}$  mol dm $^{-3}$  KCl in a dioxan/water mixture containing 30.5% water by mass). 0.05ml of PSLX-PP-1 in dioxan was added to 5ml of each of the media, to yield 45 dilute dispersions of PSLX-PP-1 (mass fraction = 0.01%). The electrophoretic mobility for each sample was measured by PALS and the corresponding zeta potentials were calculated using the Hückel equation, the Smoluchowski equation, O'Brien and White's method and Ohshima et al's method.

Table 5.11 shows the  $\kappa a$  values for each of the dispersion media, required for calculation of  $\zeta$  using O'Brien and White's or Ohshima et al's methods.

Tables 5.12, 5.13, 5.14, 5.15 and 5.15 show the electrophoretic mobilities and calculated zeta potentials for PSLX-PP-1 in water/dioxan media containing  $10^{-1}$ ,  $10^{-2}$ ,  $10^{-3}$ ,  $10^{-4}$  and  $10^{-5}$  mol dm $^{-3}$  KCl respectively. In general, the results show that for large  $\kappa a$ , the O'Brien and White method yields similar potentials to those obtained via the Smoluchowski equation, and that the Ohshima method provides a good approximation to the O'Brien and White method. It can also be seen that, as expected, the Ohshima method fails for systems

Mass fraction water/%	Electrolyte concentration				
	/mol dm <sup>-3</sup>				
	10 <sup>-1</sup>	10 <sup>-2</sup>	10 <sup>-3</sup>	10 <sup>-4</sup>	10 <sup>-5</sup>
20.0	25.1	7.94	2.51	0.794	0.251
30.5	53.1	16.8	5.31	1.68	0.531
40.5	117	37.1	11.7	3.71	1.17
49.5	216	68.4	21.6	6.84	2.16
59.8	303	95.9	30.3	9.59	3.03
69.8	318	101	31.8	10.1	3.18
79.4	386	122	38.6	12.2	3.86
89.6	360	114	36.0	11.4	3.60
100.0	331	105	33.1	10.5	3.31

Table 5.11:           Ka values for PSLX-PP-1 particles dispersed in solutions of potassium chloride in water/dioxan at 20<sup>o</sup>C as a function of dispersion medium composition.



Sample	Mobility	Calculated zeta potential/mV			
	$/10^{-10} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$	Smoluchowski	Hückel	Ohshima	O'Brien & White
A1	- 4.26 ± 1.38	- 6.9	-10.3	- 6.9	- 8.2
B1	-16.3 ± 2.6	-20.9	-31.4	-20.9	-23.0
C1	-38.2 ± 2.6	-34.2	-51.3	-34.2	-35.9
D1	-60.5 ± 2.3	-36.3	-54.4	-36.3	-37.3
E1	-82.9 ± 1.7	-37.5	-56.2	-37.5	-38.2
F1	-116 ± 3	-37.8	-56.7	-37.8	-38.6
G1	-152 ± 9	-35.4	-53.2	-35.4	-36.1
H1	-212 ± 2	-38.5	-57.8	-38.5	-39.2
I1	-278 ± 7	-35.6	-53.4	-35.6	-37.0

Table 5.12:                    Measured electrophoretic mobilities and  
calculated zeta potentials for PSLX-PP-1  
particles dispersed in water/dioxan mixtures  
containing  $10^{-1} \text{ mol dm}^{-3}$  potassium chloride  
at 25°C.

Sample	Mobility /10 <sup>-10</sup> m <sup>2</sup> s <sup>-1</sup> V <sup>-1</sup>	Calculated zeta potential/mV			
		Smoluchowski	Hückel	Ohshima	O'Brien & White
A2	-11.6 ± 0.5	-18.7	-28.0	-22.7	-23.5
B2	-38.5 ± 2.1	-49.4	-74.0	-54.1	-61.3
C2	-72.2 ± 4.3	-64.6	-96.9	-71.2	-75.9
D2	-104 ± 2	-62.3	-93.5	-66.6	-68.1
E2	-145 ± 2	-65.5	-98.3	-69.1	-70.2
F2	-190 ± 8	-61.9	-92.9	-64.9	-65.8
G2	-272 ± 3	-63.4	-95.1	-66.1	-66.9
H2	-360 ±11	-65.4	-98.1	-68.5	-69.6
I2	-386 ± 9	-49.5	-74.2	-51.5	-51.9

Table 5.13: Measured electrophoretic mobilities and calculated zeta potentials for PSLX-PP-1 particles dispersed in water/dioxan mixtures containing 10<sup>-2</sup> mol dm<sup>-3</sup> potassium chloride at 25<sup>o</sup>C.

Sample	Mobility $/10^{-10} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$	Calculated zeta potential/mV			
		Smoluchowski	Hückel	Ohshima	O'Brien & White
A3	- 21.1 ± 8	-34.0	- 51.0	-48.0	- 51.0
B3	- 60.5 ± 1.1	-77.6	-116.4	-32.6	-
C3	- 97.9 ± 0.9	-87.6	-131.4	-83.6	-
D3	-125 ± 2	-74.9	-112.4	-39.9	-115.5
E3	-167 ± 2	-75.4	-113.2	-84.7	-102.8
F3	-222 ± 1	-72.3	-108.5	-80.9	- 94.5
G3	-289 ± 3	-67.4	-101.1	-81.3	- 80.5
H3	-354 ± 4	-64.3	- 96.5	-77.3	- 77.2
I3	-459 ± 6	-58.8	- 88.2	-69.5	- 70.2

Table 5.14: Measured electrophoretic mobilities and calculated zeta potentials for PSLX-PP-1 particles dispersed in water/dioxan mixtures containing  $10^{-3} \text{ mol dm}^{-3}$  potassium chloride at  $25^{\circ}\text{C}$ .

Sample	Mobility	Calculated zeta potential/mV			
	$/10^{-10} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$	Smoluchowski	Hückel	Ohshima	O'Brien & White
A4	- 37.0 ± 0.4	-59.5	- 89.3	-15.2	-107.8
B4	- 66.4 ± 0.8	-85.1	-127.7	-28.9	-
C4	- 86.9 ± 0.3	-77.7	-116.6	-41.0	-
D4	- 93.9 ± 1.2	-56.3	- 84.4	-53.2	-100.4
E4	-122 ± 5	-55.1	- 82.7	-55.4	- 83.2
F4	-167 ± 3	-54.4	- 81.6	-55.6	- 81.2
G4	-225 ± 3	-52.5	- 78.7	-55.3	- 73.6
H4	-291 ± 4	-52.9	- 79.3	-55.1	- 76.6
I4	-331 ± 5	-42.4	- 63.6	-45.3	- 56.9

Table 5.15:      Measured electrophoretic mobilities and calculated  
                          zeta potentials for PSLX-PP-1 particles dispersed in  
                          water/dioxan mixtures containing  $10^{-4} \text{ mol dm}^{-3}$   
                          potassium chloride at 25°C.

Sample	Mobility $/10^{-10} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$	Calculated zeta potential/mV			
		Smoluchowski	Hückel	Ohshima	O'Brien & White
A5	- 34.8 ± 1.1	-56.0	- 84.0	- 6.5	- 88.7
B5	- 65.1 ± 0.6	-83.5	-125.2	-13.9	-203
C5	- 77.6 ± 0.8	-69.4	-104.1	-21.0	-206
D5	-108 ± 1	-64.7	- 97.1	-61.2	-
E5	-130 ± 5	-58.7	- 88.1	-31.9	-
F5	-157 ± 3	-51.2	- 76.7	-30.0	- 95.3
G5	-211 ± 3	-49.2	- 73.8	-31.2	- 86.1
H5	-301 ± 3	-54.7	- 82.0	-32.5	-
I5	-337 ± 5	-43.2	- 64.7	-27.5	- 71.6

Table 5.16:      Measured electrophoretic mobilities and calculated  
                          zeta potentials for PSLX-PP-1 particles dispersed in  
                          water/dioxan mixtures containing  $10^{-5} \text{ mol dm}^{-3}$   
                          potassium chloride at 25°C.

with  $\kappa a < 10$ . It is interesting to note that for systems with large  $\zeta$  and relatively small  $\kappa a$ , it is not possible to calculate  $\zeta$  using the O'Brien and White method. The reason for this is not clear.

From the data in tables 5.12 to 5.16, it is evident that the size of the maximum observed for the zeta potential as the relative permittivity of the medium changes depends strongly on the concentration of electrolyte present in the bulk phase. At high salt concentration ( $10^{-1} \text{ mol dm}^{-3}$ ), no maximum is observed. At low salt concentration, a well-pronounced maximum is evident ( $\sim 200 \text{ mV}$  using the O'Brien and White method). The reasons for the depression of the zeta potential maximum as the concentration of electrolyte increases and the rather high discrepancy between zeta potentials calculated using the O'Brien and White method and the Hückel equation at low electrolyte concentration are, as yet, unclear.

#### 5.6 MISCELLANEOUS ELECTROPHORETIC STUDIES OF NON-POLAR DRUG DISPERSIONS

In order to test the suitability of the PALS apparatus for determining the electrophoretic mobilities of drug particles dispersed in non-polar media, a variety of drugs were dispersed into trichlorotrifluoroethane (trade name "Arcton 113") containing a variety of surfactants. Five drugs were studied, designated AB, AS, SAL, BDP and FP.



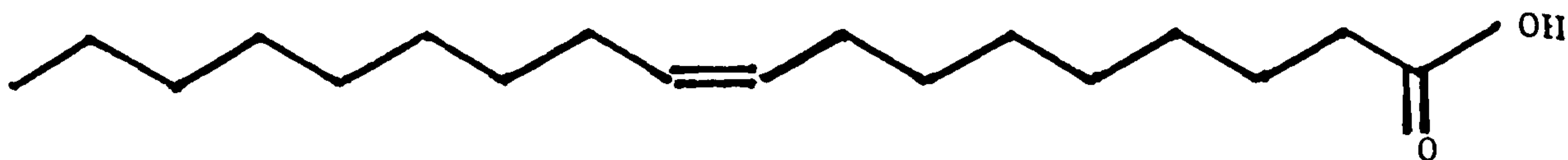
**PAGE**

**NUMBERING**

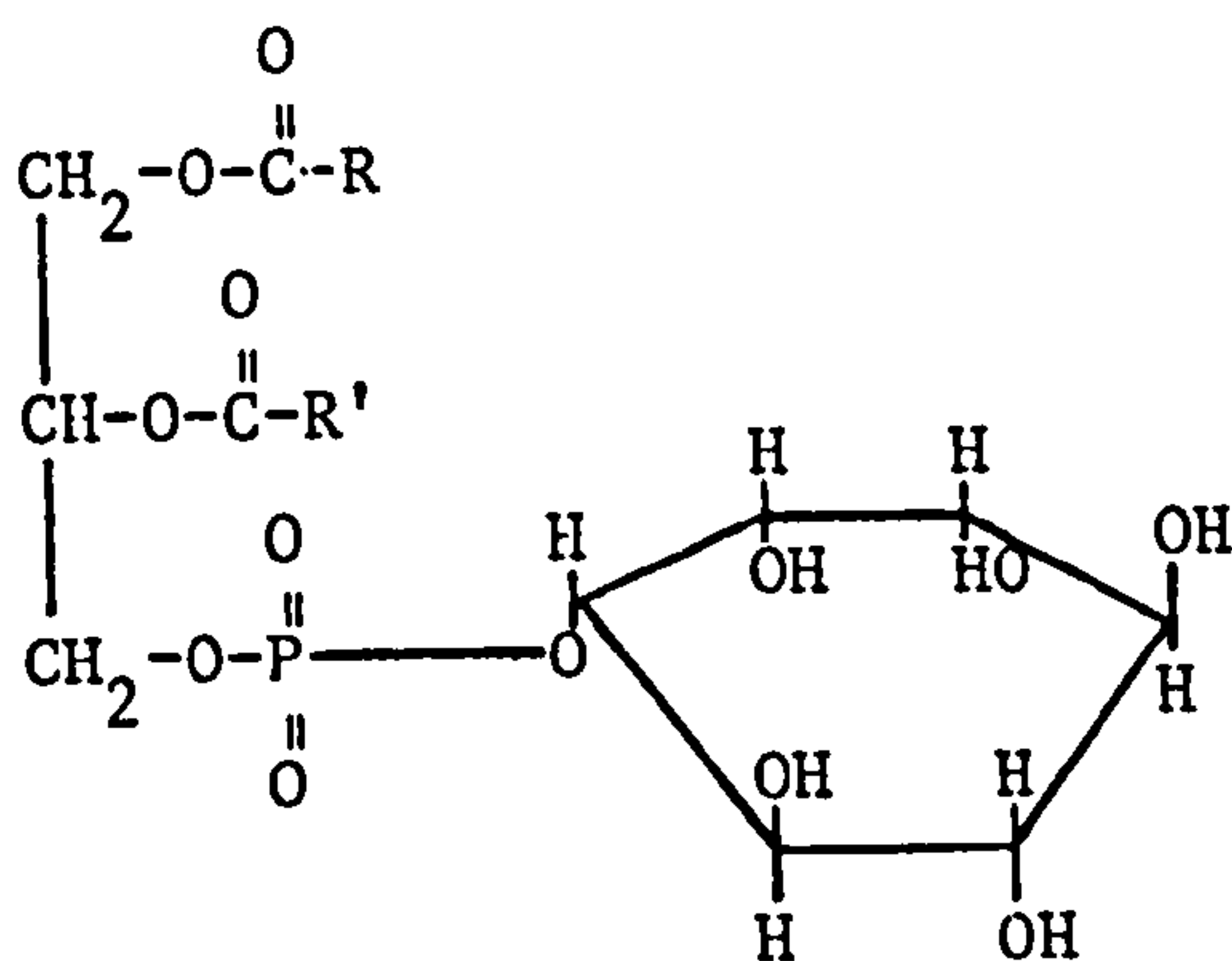
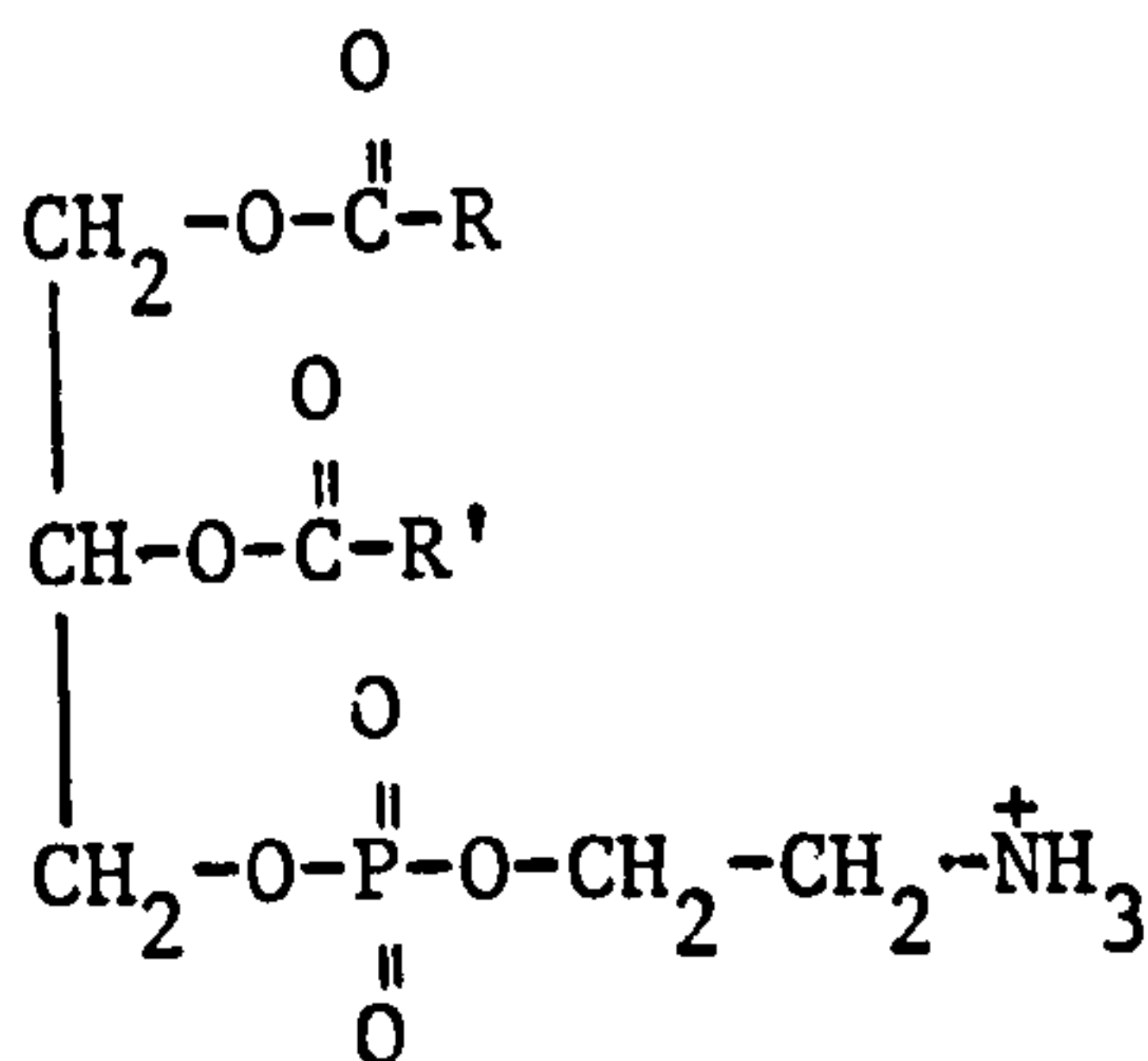
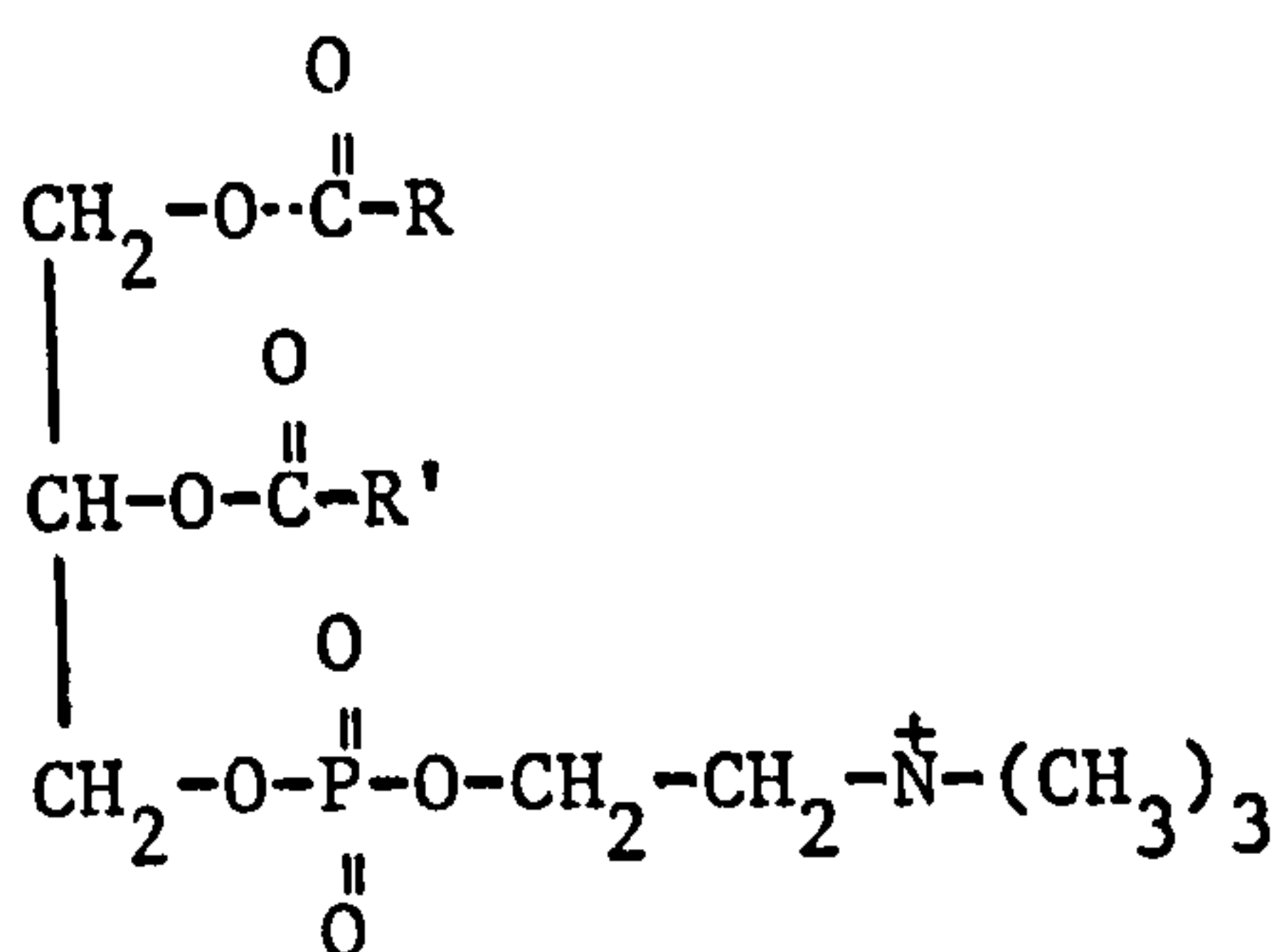
**AS ORIGINAL**

The surfactants employed were :

- (i) oleic acid (OA)

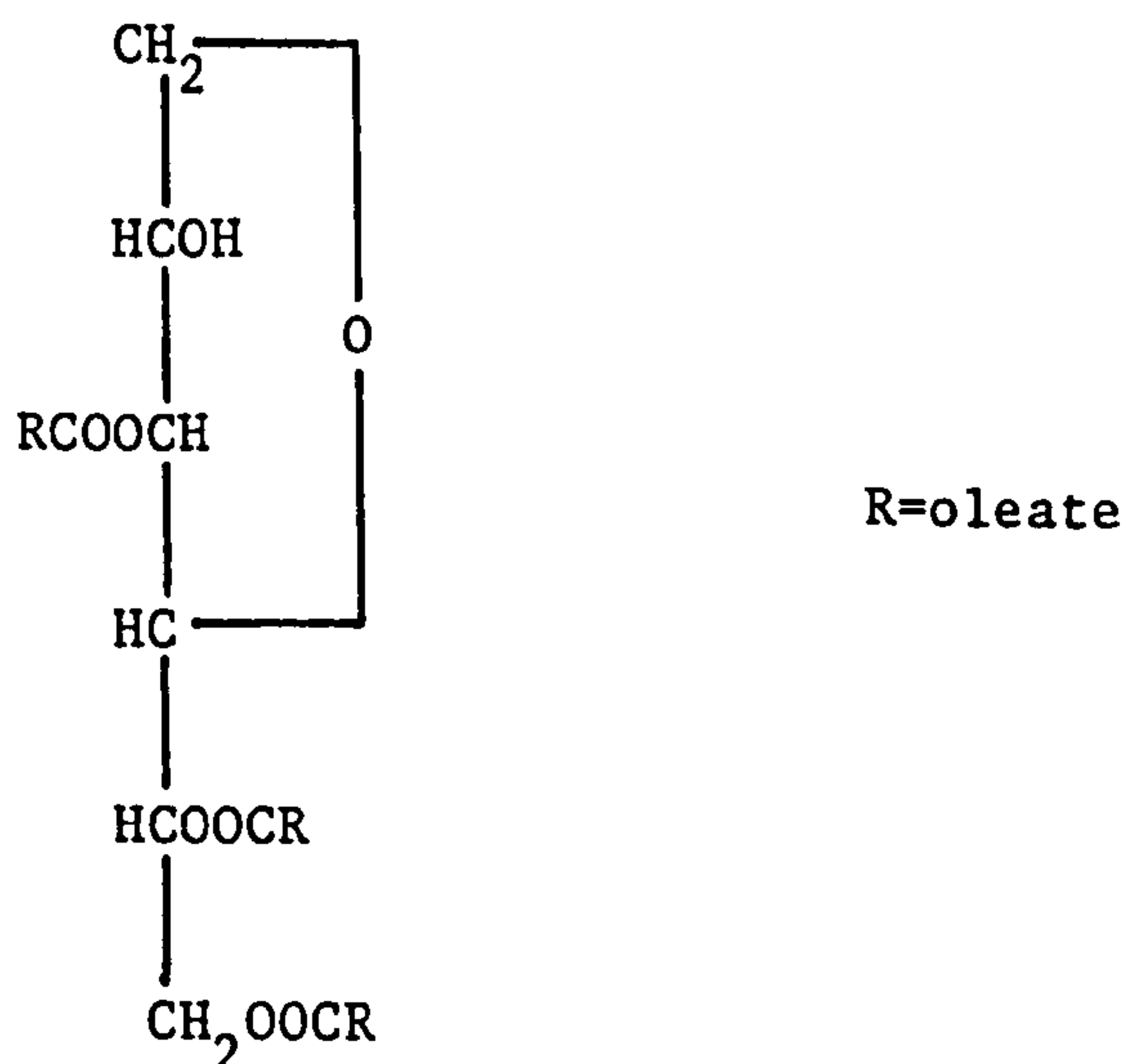


- (ii) Lecithin (LEC) - a mixture of phospholipids, containing hydrophobic and hydrophilic moieties. The main components are :



where R and R' represent fatty acid constituents.

(iii) sorbitan trioleate (ST) - the fatty acid triester of sorbitan:



The following physical properties were assumed for the Arcton 113<sup>119</sup> :

- (i) density at 25°C = 1.565 g cm<sup>-3</sup>,
- (ii) viscosity at 25°C = 0.68 x 10<sup>-3</sup> kg m<sup>-1</sup> s<sup>-1</sup>,
- (iii) relative permittivity at 25°C = 2.41.

Table 5.17 summarises the samples prepared for study by PALS. For each of the samples, 10ml of the dispersion were subjected to ultrasonification for two minutes prior to measurement using the PALS apparatus. The electrophoretic mobility was determined from the average of five experiments, each consisting of 500 samples of 50 lms -spaced data points (i.e. sampling = 5 x 500 x 50 x 0.001 s). The field employed was a 30.0 Hz sinusoidal field of amplitude 48.5V across an electrode separation of 3.93mm. The "AutoTrack" facility was employed to improve the resolution of the electrophoretic velocity information (see § 3.66 and § 4.2.13). For most of the

Drug	Mass of drug/mg	Dispersion medium	Mass of disp. medium/g	Sample code
AB	1.0	113	40.0	AB/113
AB	1.0	0.01% OA w/w	35.0	AB/OA
AB	1.0	0.01% LEC w/w	36.0	AB/OA
AB	1.0	0.01% ST w/w	35.0	AB/ST
AS	1.2	113	35.0	AS/113
AS	1.0	0.01% OA w/w	35.0	AS/OA
AS	1.0	0.01% LEC w/w	35.0	AS/LEC
AS	1.0	0.01% ST w/w	35.0	AS/ST
SAL	1.1	113	47.0	SAL/113
SAL	1.0	0.01% OA w/w	35.0	SAL/OA
SAL	1.0	0.01% LEC w/w	35.0	SAL/LEC
SAL	1.0	0.01% ST w/w	35.0	SAL/ST
BDP	2.5	113	40.0	BDP/113
BDP	5.0	0.01% OA w/w	115.0	BDP/OA
BDP	1.0	0.01% LEC w/w	35.0	BDP/LEC
BDP	1.0	0.01% ST w/w	36.0	BDP/ST
FP	1.0	113	35.0	FP/113
FP	1.0	0.01% OA w/w	35.0	FP/OA
FP	1.0	0.01% LEC w/w	35.0	FP/LEC
FP	1.0	0.01% ST w/w	35.0	FP/ST
FP	4.0	0.01% LEC w/w	112.0	FP/LEC2

Table 5.17: Details of drug dispersions prepared for study by PALS.

All surfactants were dissolved in Arcton 113. See text for definition of abbreviations.

samples, the constant velocity term was determined in the absence of an electric field and with phase demodulation occurring about  $\Delta\omega_s$  (i.e. no AutoTrack). In general, the drug particles are less dense than 113, hence creaming should be expected.

Table 5.18 summarises the electrophoretic mobilities for the various drug dispersions. The creaming velocities are shown in Table 5.19. The results show that the PALS apparatus can be successfully applied to non-polar drug dispersions, allowing good estimates of the electrophoretic mobility of the drug particles to be made, even in the presence of high constant velocity ( $> 50 \mu\text{ms}^{-1}$ ) flow in the sample, occurring through convection and/or creaming. The mobility data given in table 5.18 can only be used qualitatively. Further experiments are required in order to elucidate the mechanisms governing the influence of surfactants on the electrical properties of the drug/liquid interface.

## 5.7 ELECTROPHORETIC STUDIES OF HIGHLY VISCOUS NON-POLAR DISPERSIONS

The non-polar dispersions studied by PALS so far have consisted of bulk phases with viscosities of the order of  $10^{-3} \text{ kg m}^{-1} \text{ s}^{-1}$ . In principle, PALS should be suited for the study of samples of higher bulk viscosities, such as engine oils. With conventional LDE equipment,

Drug	Surfactant			
	113 only	OA	LEC	ST
AB	-4.83 ± 0.80	-3.89 ± 0.70	+1.10 ± 0.92	-4.66 ± 0.93
AS	+3.72 ± 0.62	-5.75 ± 0.80	-0.85 ± -.80	-0.05 ± 0.97
SAL	-0.87 ± 1.22	-0.56 ± 1.04	-3.43 ± 0.73	+0.96 ± 0.89
BDP	-2.99 ± 0.92	-3.06 ± 0.86	DISSOLVED	DISSOLVED
FP	-3.04 ± 0.85	-2.53 ± 0.30	-8.26 ± 0.58	-0.79 ± 0.31

Table 5.18: Summary of electrophoretic mobilities (in  $10^{-10} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$ )  
for the drug dispersions listed in table 5.17.

Drug	Surfactant			
	113 only	OA	LEC	ST
AB	-93 ± 12	-19 ± 2	-50 ± 5	-30 ± 4
AS	-41 ± 12	-18 ± 4	-	-
SAL	-50 ± 10	-68 ± 6	-68 ± 2	-67 ± 10
BDP	-48 ± 4	-	DISSOLVED	DISSOLVED
FP	-57 ± 19	-63 ± 11	-96 ± 38	-42 ± 2

Table 5.19: Summary of true constant velocities (in  $\mu\text{m s}^{-1}$ )  
for the drug dispersions listed in table 5.17.



it would normally be necessary to redisperse the dispersed phase of interest into a medium of lower viscosity if reliable electrophoretic mobilities are to be determined. However, this could lead to undesirable changes occurring in the surface potential due to, for example, a change in the relative Lewis acidities or basicities of the particles and the dispersion medium. If ionic surfactants are employed to stabilise the dispersion, then the change of "solvent" may alter the degree of dissociation of the surfactant, thereby possibly affecting the surface potential. The following results illustrate the importance of studying the particles of interest in the dispersion medium chosen for their application.

Calcium high base number sulphonates (CaHBNS) are colloidal dispersions of calcium hydroxide/calcium carbonate, stabilised by a surfactant layer. This layer consists of alkyl benzene sulphonates with alkyl chain lengths between C15 and C36. The overall diameter of the particles is 8 - 10 nm. The particles are dispersed in a commercial hydrocarbon solvent, "Stanco 150", which is mainly paraffinic<sup>127</sup>. CaHBNS particles are used in engine lubricating oils as a base to react with acidic residues, resulting from the combustion of fuel, in order to prevent corrosion of the engine, especially in the combustion chambers<sup>128</sup>.

Various samples of CaHBNS particles were studied by PALS as 7.5% by mass dispersions in Stanco 150 and in toluene. Table 5.20 shows how the mobilities for the particles dispersed in toluene are typically two orders of magnitude higher than for dispersions in Stanco 150, resulting from the difference in viscosity between toluene and Stanco 150. The mobilities for the dispersions in toluene were determined within five minutes of preparing the sample and again after approx. 3 hours. It is evident that the mobility is drastically reduced over this period. After a further 20 hour period, the mobility of sample 9 dispersed in toluene was found to be  $-(1.44 \pm 0.20) \times 10^{-10} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$ . Hence, over a period of approximately one day, the sign of the charge of the particle surface has reversed. Table 5.21 shows how the mobility for sample 9 dispersed in Stanco 150 remains constant over at least a one hour period. The constant velocity term is given for each mobility value to show how effective the "AutoTrack" facility can be.

Figure 5.9(a) and 5.9(b) show the AWPDP functions obtained for two of the dispersions of CaHBNS particles in Stanco 150 (Exxon codes 1 and 6 respectively). In both cases, the electric field strength was 100V across 1.00mm with a frequency of 30.0 Hz. The sampling was 5000 x 50 x 0.999 ms. Despite the rather appreciable noise in the functions, the desired sinusoidal components are easily distinguished. The calculated mobilities from these plots are  $3.95 \times 10^{-12} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$  and  $5.25 \times 10^{-12} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$ , respectively.

Exxon Sample Code	Electrophoretic mobility/ $10^{-12} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$		
	Fresh in toluene	Aged in toluene	Fresh in Stanco 150
1	586 ± 32	118 ± 10	4.18 ± 0.40
6	740 ± 10	100 ± 19	5.11 ± 0.32
9	192 ± 15	3.2 ± 31.0	9.55 ± 0.72
10	-35.9 ± 12.0	-	-1.25 ± 0.32

Table 5.20: Electrophoretic mobilities for various CaHBNS dispersions in "Stanco 150" and toluene. Mobilities for toluene dispersions measured within 5 minutes of preparing dispersion ("fresh") and typically 3 hours later ("aged").

Time	Mobility $/10^{-12} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$	Constant velocity $/\mu\text{m s}^{-1}$
0	9.94	0.87
6	9.71	-0.04
12	10.5	-0.04
17	9.75	-0.11
23	9.55	0.18
29	7.80	0.34
35	9.89	0.68
40	8.80	0.69
46	10.1	0.62
52	9.47	0.48

Table 5.21: Electrophoretic mobility and constant velocity for CaHBNS particles dispersed in Stanco 150 as a function of time. Electric field = 100.00V across 1.00mm at 30.00Hz. Sampling = 5000 x 50 x 0.999ms.

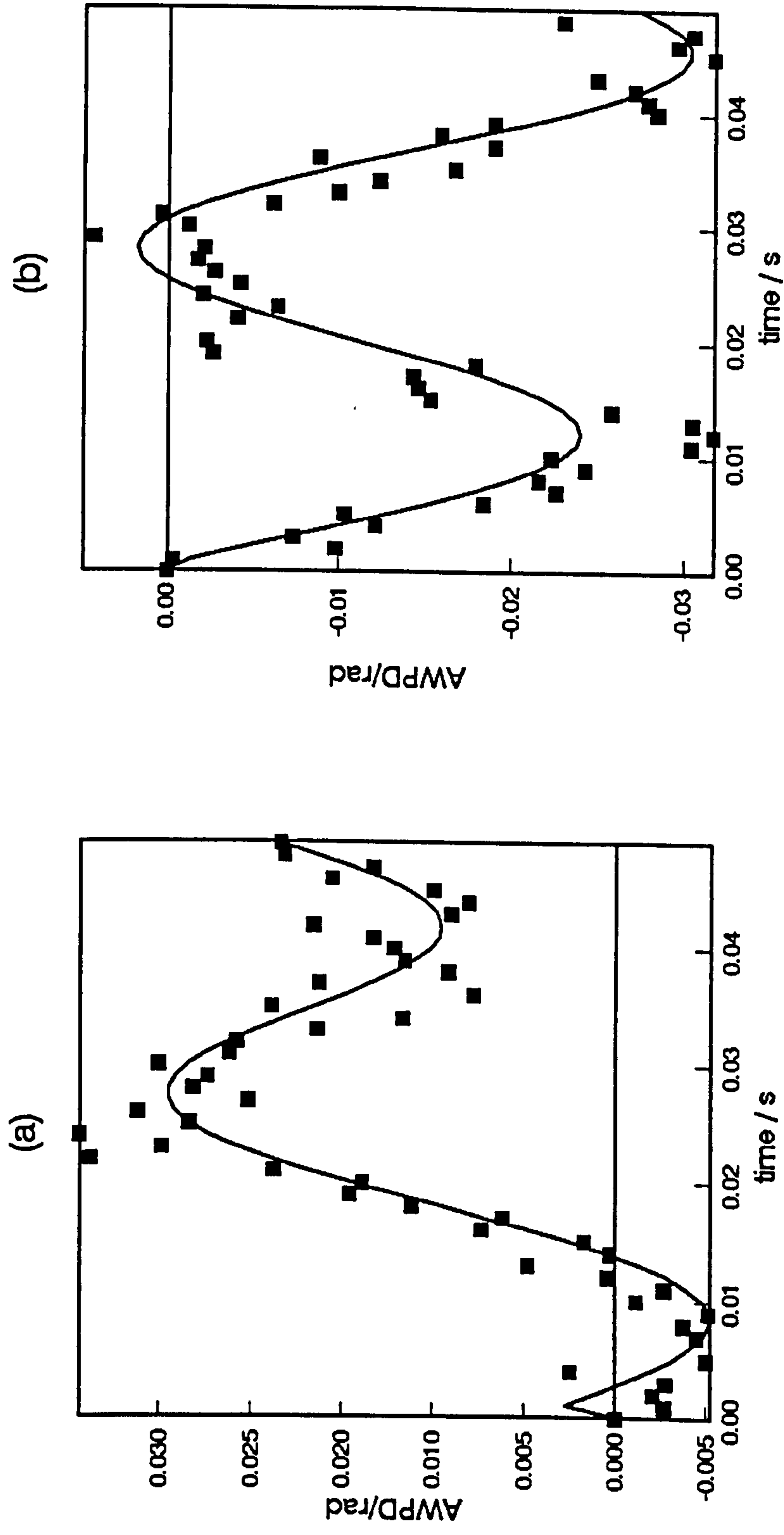


Figure 5.9: AWPD functions for two dispersions of CaHBNS particles in Stanco 150 (Exxon codes 1 and 6 for (a) and (b) respectively). Field = 100V across 1.00 mm at 30.0 Hz. Sampling = 5000 x 50 x 0.999 ms.



## CHAPTER SIX

### DISCUSSION, CONCLUSIONS AND

#### FURTHER WORK

##### 6.1 APPRAISAL OF THE PALS TECHNIQUE

This work has been concerned with the development of a technique that can allow the electrophoretic mobilities of particles dispersed in non-polar media to be measured without the need to employ high electric fields to the sample under investigation. Using the apparatus described herein, electrophoretic mobilities can be determined as low as  $\sim 10^{-12} \text{ m}^2 \text{ s}^{-1} \text{ V}^{-1}$ , with a typical statistical uncertainty of 0.5 to 5% depending on the nature of the sample. As far as the author is aware, mobilities as low as this are typically two or three orders of magnitude lower than those detectable by any other current technique.

Although PALS is based on the AWPS method of Schätzel and co-workers<sup>102, 103</sup>, the latter method only used the second moment of changes occurring in the phase of scattered light from a classical real fringe laser-Doppler electrophoresis experiment to estimate velocity information. PALS uses the first moment of the phase change to estimate electrophoretic velocities and, hence, is inherently



more sensitive. Random contributions to the phase information, such as due to diffusion, average to zero and constant velocity terms, such as due to convection, can be minimised.

PALS allows very small electrophoretic mobilities to be measured within one or two minutes for most samples. Mobilities can be determined for samples which are otherwise difficult to study using classical methods. It has been seen that particle size estimates can be made in two ways at the same time as determining electrophoretic mobilities.

In contrast to classical, static fringe LDE techniques, where it is important to optimise the particle size to fringe spacing ratio and the particle concentration in the sample, PALS only requires that the scattered light be detected; i.e. as long as a sample scatters light, PALS measurements should be possible for that sample. This means, that, as for PCS, the lower size limit for particles being studied is only limited by the power of the laser used in the experiment. The upper size limit is not limited by sedimentation, which means, in practical terms, that particles of the order of a few microns in size may be studied by PALS. It has been shown that a variety of drug particles of this size range dispersed in non-polar media can be successfully studied using PALS.

One disadvantage of the PALS technique is that one can only determine average mobilities. Hence, a dispersion containing two electrokinetically distinct species will yield a single mobility value, whereas classical spectral analysis yields a spectrum of mobilities for the sample under study. However, most systems studied usually consist of one species of particle and, hence, this apparent disadvantage can be ignored.

The speed, sensitivity and versatility of PALS means that it is easy to confirm the presence or absence of significant electrostatic repulsion contributions in a wide variety of non-aqueous dispersions, where it is thought that only steric repulsions occur.

## 6.2 SUGGESTIONS FOR FURTHER WORK

The study of the effect of trace amounts of water on the electrophoretic mobility of cross-linked polystyrene microgel particles dispersed in toluene (§ 5.3) shows that the PALS technique can be successfully applied to the study of such systems. However, it is apparent that the method of preparing the samples can influence the electrophoretic mobility through contamination of the sample by soluble pigments in the rubber septa used to seal the samples from the atmosphere. In order to carry out measurements of such non-polar

systems, it is desirable to use a dry box for sample preparation. However, such a facility was not available during this period of work. In addition to a dry box, a new sample chamber could be employed which would allow dry samples to be transferred to an air tight measurement chamber within the dry box. This chamber could then be transferred to the PALS apparatus in order to study the sample and then returned to the dry box afterwards. By incorporating the Karl Fischer titration apparatus in the dry box, the water content of the sample could be determined after study by PALS without the sample having to be exposed to the atmosphere. A sample chamber that could withstand pressures up to ~ 10 atmospheres would allow pressurised dispersions to be studied, such as found in the "Ventolin" system (see Introduction).

It would be of interest to investigate the polystyrene microgel/toluene systems further, with particular emphasis on the influence on the electrophoretic mobility of particle size, degree of cross-linking, nature of surface ionogenic groups and amount of trace water present in the dispersions.

The polystyrene microgel/dioxane/water/electrolyte system (§ 5.5) could be further studied by determining stability ratios for the samples and comparing the experimental value with that predicted by DLVO theory. However, the latter requires that the zeta potential

be determined from the electrophoretic mobility. As has been seen, there appears to be some discrepancy between the various methods used to calculate the zeta potential.

The drug systems investigated in § 5.6 should be studied further, with particular regard to the effect of the concentration of surfactant and trace water on the electrophoretic mobility and stability of the dispersions. The applicability of DLVO theory to this type of system could then be tested.



## APPENDIX 1

### THE STATISTICAL ACCURACY OF AWPD AND AWPS FUNCTIONS<sup>104</sup>

Consideration of the statistical accuracy of AWPD and AWPS functions enables the experimenter to optimise the signal-to-noise performance of a PALS experiment. In the following estimation of signal-to-noise ratios, detection noise is considered negligible compared with random fluctuations of the Doppler signal due to particle diffusion. This assumption is valid for most PALS experiments.

Let  $T$  be the total measurement time of the experiment, which comprises  $M$  cycles of the field period,

$$T = 2 \pi M / \omega. \quad (\text{A1.1})$$

and  $t_s$  the sampling time, which is simply the interval between successive data points in the AWPD/AWPS functions. For simplicity, it is assumed that the field period is an exact multiple of the sampling time,

$$L t_s = 2 \pi / \omega. \quad (\text{A1.2})$$

Then our two estimators may be written as

$$\text{AWPD} (t) = (1/M) \sum_{j=1}^M [Q(t + t_j) - Q(t_j)] \quad (\text{A1.3})$$

with

$$t_j = 2 \pi j / \omega = j L t_s, \quad (\text{A1.4})$$

which is the start time of cycle  $j$  of the field, and

$$\text{AWPS}(t) = (1/LM) \sum_{j=1}^{LM} [Q(t + jt_s) - Q(jt_s)]^2. \quad (\text{A1.5})$$

The time displacement,  $t$ , will typically be restricted to some positive integer multiple of  $t_s$ ,

$$t = k t_s. \quad (\text{A1.6})$$

The amplitude weighted phase,  $Q(t)$ , may again be separated into collective and diffusive contributions. In the most interesting case of strong diffusion, the latter term will dominate the fluctuations of our two estimators. Hence, we will restrict our attention to fluctuations of the diffusional contributions only.

The variance of  $Q^{(d)}(t + t') - Q^{(d)}(t')$  has already been calculated in eq. (3.100). Hence, we readily obtain an estimate for the noise on the AWPD given by



$$\begin{aligned}
\delta_{\text{AWPD}}(t) &\approx \left\{ (1/M) \langle [Q^{(d)}(t + t') - Q^{(d)}(t')]^2 \rangle \right\}^{0.5} \\
&= \left[ \langle A^2 \rangle q^2 D t / M \right]^{0.5} \\
&= (4 / \pi) \langle A \rangle q (2 \pi / \omega) [D k / L T]^{0.5} \\
&= 8 (\langle A \rangle q / \omega) [D k / L T]^{0.5} . \tag{A1.7}
\end{aligned}$$

The prefactor  $\langle A \rangle q / \omega$  has already occurred in the expectation of the AWPD.  $k$  will typically vary between 0 and  $L$ . Hence, the relative noise on the AWPD will be of the order

$$\delta_{\text{AWPD}}(t) / \text{AWPD}(t) \approx 8 (D/T)^{0.5} / (\mu E_0) . \tag{A1.8}$$

It is important to note that this relative noise expression does not depend on the field frequency,  $\omega$ . Hence, the signal-to-noise performance of the AWPD measurement does not degrade with increasing field frequency. Thus, comparatively high field frequencies are permitted, which will reduce problems such as electrolysis and electroosmotic flow.

A practical upper limit for the field frequency results from the requirement to decrease the sampling time in proportion to the increase in field frequency. At some stage, this sampling time will approach hardware limits or the larger detection bandwidth will lead to problems with detector noise.

Before calculating the variance of the AWPS estimator, let us consider its expectation. Both quantities are most easily determined by breaking the Q-differences up like

$$Q(t + jt_s) - Q(jt_s) = \sum_{l=1}^k \Delta Q_{j+l} \quad (\text{A1.9})$$

for  $t = kt_s$  and

$$\Delta Q_m = Q(mt_s) - Q((m-1)t_s). \quad (\text{A1.10})$$

The  $\Delta Q_m$  are then independent Gaussian random variables with mean 0 and variance

$$\langle \Delta Q_m^2 \rangle = \langle A^2 \rangle q^2 D t_s,$$

obtained from eq. (3.100) for the limit of electrophoretic motion small compared to diffusion. With this notation, the expectation of the AWPS estimator is

$$\begin{aligned} \langle \text{AWPS}(t) \rangle &= (1/LM) \sum_{j=1}^{LM} \left\langle \left( \sum_{l=1}^k \Delta Q_{j+l} \right)^2 \right\rangle \\ &= (1/LM) \sum_{j=1}^{LM} \sum_{l=1}^k \sum_{m=1}^k \langle \Delta Q_{j+l} \Delta Q_{j+m} \rangle \\ &= k \langle \Delta Q_m^2 \rangle = \langle A^2 \rangle q^2 D t, \end{aligned} \quad (\text{A1.11})$$

which is the desired value for vanishing collective particle motion. The calculation of the estimator variance proceeds along the same lines, but involves a six-fold summation over fourth moments of the  $\Delta Q_m$ . Only terms like  $\langle \Delta Q_m^2 \rangle^2$  and  $\langle \Delta Q_m^4 \rangle = 3 \langle \Delta Q_m^2 \rangle^2$  will survive the averaging. This fact will considerably reduce the number of relevant summation terms.

$$\begin{aligned}
 \text{Var (AWPS)} &= (1/LM)^2 \sum_{j=1}^{LM} \sum_{l=1}^k \sum_{m=1}^k \sum_{n=1}^{LM} \sum_{p=1}^k \sum_{q=1}^k \{ \\
 &\langle \Delta Q_{j+1} \Delta Q_{j+m} \Delta Q_{n+p} \Delta Q_{n+q} \rangle - \langle \Delta Q_{j+1} \Delta Q_{j+m} \Delta Q_{n+p} \Delta Q_{n+q} \rangle \} \\
 &= (1/LM)^2 LM k^2 (\langle \Delta Q_m^4 \rangle - \langle \Delta Q_m^2 \rangle^2) + \frac{2}{3} LM K (2k - 1) (k - 1) \langle \Delta Q_m^2 \rangle^2 \\
 &= (2/3LM) (2k^2 + 1) k (\langle A^2 \rangle q^2 D t_s)^2 . \tag{A1.12}
 \end{aligned}$$

For comparison with the AWPDP, it is necessary to consider the ratio of the square root of this variance to the expectation of the electrophoretic contribution to the AWPS, as obtained in eq. (3.102).

For simplicity,  $k=L$ , which leads to an inverse signal-to-noise ratio

$$\begin{aligned}
\delta_{\text{AWPS}} / \langle \Delta Q^{(c)} (L t_s)^2 \rangle &= \left( \frac{2}{3LM} (21^2 + 1)1 \right)^{0.5} \frac{4}{\pi} D t_s \omega^2 / (\mu^2 E_o^2) \\
&= 16/\sqrt{3} (t_s/T)^{0.5} (D\omega/\mu^2 E_o^2) \\
&= \left[ 8 (D/T)^{0.5} / (\mu E_o) \right] \cdot \frac{2}{\sqrt{3}} (D t_s)^{0.5} (\omega/\mu E_o).
\end{aligned}$$

(A1.13)

Since it was assumed that diffusion dominates, the noise on the AWPS is typically much larger than on the AWPD. For large  $\mu E_o/\omega$ , the above derivations would need to include additional collective motion contributions in the variance expression, eq. (A1.12).

## APPENDIX 2

### SOFTWARE LISTING AND DESCRIPTION

The following pages list the source code files for the software developed for the PALS apparatus. The software is written in Turbo Pascal 4.0, except for "5210 DATA.BAS", which is written in Microsoft QuickBASIC 3.0. A general knowledge of Pascal, BASIC and MS-DOS Version 3.3 is assumed. A description of the important aspects of the software follows the listings. Programming details for the hardware controlled by the software can be found in the relevant manuals.



## LISTING FOR FILE C:\PASCAL\WORK\PALES24.PAS

```

1:  Program MasterAUPS;
2:
3:  {-----}
4:  {-      Phase Analysis Light Scattering Master Program Kernel      -}
5:  {-                                                                 -}
6:  {-      Version 2.4          27th March 1990          -}
7:  {-                                                                 -}
8:  {-      by John F. Miller          -}
9:  {-                                                                 -}
10: {-      Department of Physical Chemistry          -}
11: {-      University of Bristol          -}
12: {-      Cantock's Close          -}
13: {-      Bristol          -}
14: {-      BS8 1TS          -}
15: {------}
16:
17: {------}
18: {- Developed from MasterAUPS Version 22.0    9 Aug 89          -}
19: {- Residual bugs removed 28/11/89          --> PALS Version 1.0 -}
20: {- Adapted for using 4-way electrodes 9/1/90 --> PALS Version 2.0 -}
21: {-      Revised 25/1/90 --> PALS Version 2.1          -}
22: {- 'AutoTrack' facility incorporated 25/1/90 --> PALS Version 2.2 -}
23: {- Autocorrelation of real component used to          -}
24: {- calculate Dq2 info. incorporated 16/1/90 --> PALS Version 2.3 -}
25: {- Electrode voltage autoswitchover unit installed: If input is > 10V, the -}
26: {- output is the input divided by 10, else it is unaltered. A TTL output is -}
27: {- provided so the computer can automatically compensate for the division. -}
28: {-      Incorporated 27/3/90 --> PALS Version 2.4          -}
29: {-                                                                 -}
30: {- for use with PARC 5210 Dual-Phase Lock-In Amplifier which needs to be -}
31: {- connected to the Vectra via the COM1 RS-232C port. The 5210 RS-232C -}
32: {- interface should be preconfigured with parameters 0 = 11 and 1 = 16. See -}
33: {- 5210 manual for further details. The delimiter character should be SPACE.-}
34: {-                                                                 -}
35: {- Real-time processing version - uses double-buffered data capture and -}
36: {- integer arithmetic for creating the various functions. Two 254 x 254 8bit-}
37: {- look-up tables are used for determining the amplitude and phase values -}
38: {- from the 2 analogue inputs. The real-time processing is available down to-}
39: {- sampling times of 1ms. (0.5ms may be obtained if correlation/squared AUPS-}
40: {- functions are not calculated.)          -}
41: {- Non-squaring version...data files not compatible with pre 21.0 versions. -}
42: {- i.e. AUPD and AUPS functions constructed.          -}
43: {- Multiple run version....allows for multiple runs with given parameter -}
44: {- setup. Mobility etc. data may be written to printer or disc.          -}
45: {- Sign of mobility is determined using second derivative of AUPS fit. -}
46: {- Diffusion coefficient estimated via amplitude correlation function. -}
47: {- Short form files include average and standard deviation calculations. -}
48: {- F1 function key redefined for each i/p request.          -}
49: {- Determination of linear term in AUPS function is improved by subtracting -}
50: {- away the constant velocity term, obtained from AUPD function.          -}
51: {- 'AutoTrack' facility available from MultiRun menu (25/1/90) - external -}
52: {- frequency generator can be adjusted via a voltage from D/A channel 0 of -}
53: {- the DT2818. The software calibrates the voltage with the frequency shift-}
54: {- by measuring the 5210 reference frequency (Data.LIAReffreq) at -0.100V & -}
55: {- +0.100V. After an experiment, the voltage is adjusted to shift the freq -}

```



```

56: {- by  $A/2\pi$  Hz, where A is the linear term of the AWPD function. The menu -}
57: {- operations appear in the MultiRun menu: -}
58: {- -}
59: {-          5. Enable AutoTrack (Currently OFF) -}
60: {- or:      5. Disable AutoTrack (Currently ON) -}
61: {-          6. Calibrate AutoTrack -}
62: {-          7. Main Menu -}
63: {- -}
64: {- AutoTrack is off when the program is first started and must be calibrated-}
65: {- before it is used. -}
66: {- -}
67: {- Aborting an experiment: -}
68: {- -}
69: {- The current experiment can be aborted by pressing <CTRL>-X once. For a -}
70: {- series of runs (ie MultiRun), the software will process the data so far -}
71: {- obtained and the proceed with the next run. To abort a series of runs, -}
72: {- press <CTRL>-X twice in rapid succession. -}
73: {- -}
74: {- An experiment will abort automatically if DMA transfer timeout occurs, as-}
75: {- will happen if the electrode drive is turned off during the run. If the -}
76: {- electrode drive is off before the run starts, but should be on, an error -}
77: {- message will be displayed asking the operator to make sure it is on. -}
78: {- -}
79: {- 'Grapher' compatible results files: -}
80: {- -}
81: {- As well as long form, binary data files, the results may be written as an-}
82: {- ASCII file for use by, primarily, Grapher. The amplitude histogram, AWPD-}
83: {- AUPS and amp. correlation functions are written as a function of time. -}
84: {- The necessary Grapher instructions to generate axis labels, titles etc. -}
85: {- are written. The file should be created after determining the mobility. -}
86: {- The files may also be imported into 123 or Freelance. -}
87: {------}
88:
89:
90: {$R-,S-,I-,O+,I+,F-,V+,B-,H+,L+ }
91: {$M 32768,0,150000 }
92:
93: {- This is the main AUPS program.... -}
94:
95: Uses DT2818, AUPSUnit,
96:     {$U LeastSq4.PAS } LeastSq, Graph, Derivs, Crt, Dos;
97:
98:
99: {- Routines that are regularly edited in some way are stored as INCLUDE -}
100: {- files and are located in the Turbo PASCAL 'INC' directory as specified -}
101: {- in the Options menu. Routines that require very little editing are -}
102: {- stored in various units ('UNIT' directory) such as DT2818, which defines-}
103: {- the low level structures required to operate the DT2818 Simultaneous -}
104: {- Sample and Hold Analog/Digital board. -}
105:
106: {- Documentation for the include files will be found within each file. -}
107:
108:
109: {$I DEF_VARS.INC}
110: {$I PROCS_1.INC}

```

```

111:  {$I 5210DATA.INC}
112:  {$I STRFUNCS.INC}
113:  {$I MENUS24.INC}
114:  {$I INPUTS24.INC}
115:  {$I ENTMENUS.INC}
116:  {$I FITAWPS2.INC}
117:  {$I VIEWSIG5.INC}
118:
119:  Procedure InitGraphics;
120:
121:  {- Invoke graphics handling software to drive CGA board in 640x200 mode  -}
122:
123:  Var
124:      GraphDriver : Integer;
125:      GraphMode   : Integer;
126:
127:  Begin
128:      GraphDriver := CGA;
129:      GraphMode  := CGAHi;
130:      InitGraph(GraphDriver, GraphMode, '');
131:      TextMode(3);
132:  End; {- of InitGraphics -}
133:
134:  {$I RUNEXP.INC}
135:  {$I DOSPROCS.INC}
136:  {$I MULTIRUN.INC}
137:  {$I ENTMENU2.INC}
138:
139:  Procedure InitialiseVariables;
140:
141:  {- Read in a set of typical parameters for start-up so that menu displays -}
142:  {- do not display nonsensical information. By default, the parameter data -}
143:  {- will be read from 'C:\AWPS\PARAMS.FIL'. If another file is required, -}
144:  {- this can be specified when calling the software from DOS, i.e: -}
145:  {- -}
146:  {-          AWPS /P C:\PVDC2_1.5 . -}
147:  {- -}
148:  {- will read in parameters from file 'PVDC2_1.5' from drive C: -}
149:
150:  Var
151:      Paramfile : String;
152:
153:  Begin
154:      If (ParamStr(1) = '/p') or (ParamStr(1) = '/P') then
155:          Paramfile := ParamStr(2)
156:      else
157:          Paramfile := 'C:\AWPS\PARAMS.FIL';
158:      Assign(Datafile, Paramfile);
159:      Reset(Datafile);
160:      If IOResult <> 0 then begin
161:          UserPrompt(Paramfile+' not found. Press SPACEBAR. ');
162:          Repeat Until ReadKey = ' ';
163:          InitialiseVariables;
164:      end
165:      else

```

```

166:         begin
167:             UserPrompt('Reading parameter data from '+ParamFile);
168:             Read(DataFile, Data);
169:             Close(DataFile);
170:         end;
171:         LongName := 'NUL';
172:         ShortName := 'NUL';
173:         NumRuns := 1;
174:         DataFileName := '';
175:         CurrentVolume := '';
176:         AutoTrack := False;
177:         CentreOffset := 0;
178:     end; {- of initialise variables -}
179:
180:     Begin
181:
182:         {- Reroute input via DOS rather than Crt to enable key redefinitions etc -}
183:
184:         Assign(Input, ''); Reset(Input);
185:         InitialiseVariables;
186:         GetDir(0, CurrentDirectory);
187:         ClearUserPrompt;
188:         CalibrateAutoTrack;
189:
190:         {- Check that the CGA graphics driver has been successful linked during -}
191:         {- compilation. If not, program will terminate. -}
192:
193:         If RegisterBGIDriver(@CGADriver) <> 0 then
194:             begin
195:                 Writeln('Graphics error: Driver not installed');
196:                 Halt;
197:             end;
198:         InitGraphics;
199:
200:         {- This is the main menu loop and is self-explanatory. -}
201:
202:         Repeat Begin
203:             DisplayMainMenu;
204:             Case ReadKey of
205:                 '1' : EnterParameterMenu;
206:                 '2' : EnterDataStorageMenu;
207:                 '3' : ViewPSDSignals;
208:                 '4' : EnterDataProcessingMenu;
209:                 '5' : RunExperiment(Prompts, 1);
210:                 '6' : EnterMultiRunMenu;
211:                 '7' : Begin
212:                     UserPrompt('Are you sure you want to quit? (Y to confirm)');
213:                     Case ReadKey of
214:                         'Y' : Halt;
215:                         'y' : Halt;
216:                     End;
217:                     End;
218:                 '8' : Begin
219:                     UserPrompt('Type EXIT to return to PALS environment...');
220:                     Exec('C:\COMMAND.COM', '');

```

```
221:         GetDir(0, CurrentDirectory);
222:     End;
223:     else
224:         ErrorBeep;
225:     end;
226:     end;
227: Until False;
228: End.
```

## LISTING FOR FILE C:\PASCAL\INC\DEF\_VARS.INC

```

1:  (- DEF_VARS.INC - 5210 VERSION -)
2:  (- Define types and constants -)
3:
4:  {$L CGADRV.OBJ }
5:
6:  Procedure CGADriver; external;  (- Link CGA driver to main program -)
7:
8:  Const
9:      LensApertureStr : array[1..9] of string[8] = ('f3.8','f5.6',
10:                                                    'f5.6/8','f8','f8/11','f11',
11:                                                    'f11/16','f16','f22');
12:      PinHoleSizeStr  : array[1..3] of string[6] = ('None','200 um','400 um');
13:      LensTubeStr     : array[1..8] of string[8] = ('None','10 mm','20 mm','30 mm',
14:                                                    '10+20 mm','10+30 mm','20+30 mm',
15:                                                    'All');
16:      LIASensStr      : array[0..15] of string[6] = ('100 nV','300 nV','1uV','3uV',
17:                                                    '10uV','30uV','100uV','300uV','1mV',
18:                                                    '3mV','10mV','30mV','100mV',
19:                                                    '300mV','1V','3V');
20:      LIAFilterICStr  : array[0..13] of string[6] = ('1ms','3ms','10 ms','30 ms',
21:                                                    '100 ms','300ms','1 s','3 s',
22:                                                    '10 s','30 s','100 s','300 s',
23:                                                    '1000 s','3000 s');
24:      LIAFilterTuningModeStr : array[0..1] of string[6] = ('MANUAL','TRACK');
25:      LIADynamicReserveStr : array[0..2] of string[7] = ('HI STAB','NORM','HI RES');
26:      LIAFilterModeStr : array[0..3] of string[9] = ('FLAT','NOTCH','LOW-PASS',
27:                                                    'BAND-PASS');
28:      LIALineFiltersStr : array[0..3] of string[6] = ('NONE','100 Hz','50 Hz','BOTH');
29:      LIAFilterRollOffStr : array[0..1] of string[5] = ('6 dB','12 dB');
30:      NoPrompts : boolean = false;
31:      Prompts : boolean = true;
32:
33:  Type
34:      LensApertureType      = 1..9;
35:      PinHoleSizeType       = 1..3;
36:      LensTubeType         = 1..8;
37:      LIASensType           = 0..15;
38:      LIAFilterICType       = 0..13;
39:      LIAFilterTuningModeType = 0..1;
40:      LIADynamicReserveType = 0..2;
41:      LIAFilterModeType     = 0..3;
42:      LIALineFiltersType    = 0..3;
43:      LIAFilterRollOffType  = 0..1;
44:
45:
46:  (- DataRecord defines the format of the experimental data as stored on disc-)
47:  (- (Note: Stored data is stored as a 'binary' file and hence cannot be read-)
48:  (- directly from DOS. A PASCAL program should be written if one wishes to -)
49:  (- access data files outside of this program.) -)
50:
51:  DataRecord = record
52:      Wavelength, Angle, DelayTime : extended;
53:      LensAperture : LensApertureType;
54:      PinHoleSize : PinHoleSizeType;
55:      LensTubes : LensTubeType;

```



```

56:      CellType : String[20];
57:      PMSupply : word;
58:      LIAFilterTuningMode : LIAFilterTuningModeType;
59:      LIADynamicReserve : LIADynamicReserveType;
60:      LIAFilterMode : LIAFilterModeType;
61:      LIAReffreq : LongInt;
62:      LIAVernierGainStatus : Byte;
63:      LIAVernierGain : Byte;
64:      LIALinefilters : LIALinefiltersType;
65:      LIAOscAmplitude : Word;
66:      LIAOscfreq : Word;
67:      LIAOscfreqBand : Byte;
68:      LIASens : LIASensType;
69:      LIAFilterIC : LIAFilterICType;
70:      LIAFilterRollOff : LIAFilterRollOffType;
71:      Electrodefreq, ElectrodeAmp, ElectrodeGap : extended;
72:      ElectrodeType : String[20];
73:      Viscosity, RefIndex, Temperature : extended;
74:      SampleName : String[30];
75:      Date, Time : String[10];
76:      NumPoints, NumSamples : Word;
77:      RejectionFactor : extended;
78:      AWPData, TimeData, Histogram, AmpCorr, AWPData : DerivArray;
79:      AmpCorrelationFactor : extended;
80:  end;
81:
82:  Var
83:      DataFile : File of DataRecord;
84:      Data : DataRecord;
85:      CurrentVolume, CurrentDirectory, DataFileName : String;
86:      GrError : integer;
87:      GlobalIntercept, GlobalGradient, GlobalSN : extended;
88:      GlobalMobility, GlobalVelocity, GlobalDiameter, GlobalSD : extended;
89:      GlobalATerm : extended;
90:      AutoTrack : boolean;
91:      CentreOffsetInc, freqToVoltsFactor : extended;
92:      NumRuns : word;
93:      LongName, ShortName : String;
94:
95:  {- End of DEF_VARS.INC -}

```



## LISTING FOR FILE C:\PASCAL\INC\PROCS\_1.INC

```

1:  {- This include file (PROCS_1.INC) contains some miscellaneous procedures -}
2:  {- and functions.                                     -}
3:
4:  Procedure ErrorBeep;
5:
6:  {- This procedure emits a small beep, usually in response to operator error-}
7:  {- or invalid disc operation.                                     -}
8:
9:  Begin
10:   Sound(440);
11:   Delay(500);
12:   NoSound;
13: End; {- of ErrorBeep -}
14:
15: Function CheckforIOError : Boolean;
16:
17: {- Following disc input/output, this function is called to check that the -}
18: {- operation was successful. If not, the appropriate error message is -}
19: {- displayed at the bottom of the screen.                                     -}
20:
21: Var
22:   IOError : Word;
23:   IOErrorStr : String;
24:
25: Begin
26:   IOError := IOResult;
27:   If IOError = 0 then Begin
28:     CheckforIOError := False;
29:     Exit;
30:   End;
31:   CheckforIOError := True;
32:   ErrorBeep;
33:   Case IOError of
34:     2 : IOErrorStr := 'File not found. ';
35:     3 : IOErrorStr := 'Path not found. ';
36:     4 : IOErrorStr := 'Too many open files. ';
37:     5 : IOErrorStr := 'File access denied. ';
38:     6 : IOErrorStr := 'Invalid file handle. ';
39:     12 : IOErrorStr := 'Invalid file access code. ';
40:     15 : IOErrorStr := 'Invalid drive number. ';
41:     16 : IOErrorStr := 'Cannot remove current directory. ';
42:     17 : IOErrorStr := 'Cannot rename across drives. ';
43:     100 : IOErrorStr := 'Disc read error. ';
44:     101 : IOErrorStr := 'Disc write error. ';
45:     102 : IOErrorStr := 'File not assigned. ';
46:     103 : IOErrorStr := 'File not open. ';
47:     104 : IOErrorStr := 'File not open for input. ';
48:     105 : IOErrorStr := 'File not open for output. ';
49:     106 : IOErrorStr := 'Invalid numeric format. ';
50:   Else
51:     IOErrorStr := 'Undefined error. ';
52:   End;
53:   IOErrorStr := 'IOError: '+IOErrorStr+'Press SPACEBAR';
54:   TextColor(Yellow);
55:   TextBackGround(Red);

```

```

56:   GotoXY(1,25);
57:   Write(IOErrorStr);
58:   TextBackGround(Black);
59:   Repeat Until ReadKey = ' ';
60:   End; {- of CheckforIOError -}
61:
62:   Procedure DefFuncKey(DefinitionStr : String);
63:
64:   {- This procedure redefines function key f1 as 'String'           -}
65:
66:   Var f : text;
67:
68:   Begin
69:     Assign(f, '');
70:     Rewrite(f);
71:     Write(f, Chr(27)+'[O;59;"'+DefinitionStr+'"p');
72:     Close(f);
73:   End; {- of DefFuncKey -}
74:
75:
76:   Procedure DefFuncKeyExtended(Definition : Extended;
77:                                Width, Decimals : Word);
78:
79:   {- This procedure redefines function key f1 with the string representation -}
80:   {- of the extended variable 'Definition' with minimum width 'Width' and      -}
81:   {- no. of decimal 'Decimals'.                                                -}
82:
83:   Var DefStr : String;
84:
85:   Begin
86:     Str(Definition:Width:Decimals, DefStr);
87:     DefFuncKey(DefStr);
88:   End; {- of DefFuncKeyExtended -}
89:
90:   Procedure DefFuncKeyWord(Definition : Word);
91:
92:   {- This procedure redefines function key f1 with the string representation -}
93:   {- of the word variable 'Definition'.                                         -}
94:
95:   Var DefStr : String;
96:
97:   Begin
98:     Str(Definition, DefStr);
99:     DefFuncKey(DefStr);
100:   End; {- of DefFuncKeyWord -}
101:
102:
103:   {- End of PROCS_1.INC -}

```

## LISTING FOR FILE C:\PASCAL\INC\5210DATA.INC

```

1:  {- This file (5210DATA.INC) defines the procedure to fetch the 5210 setup  -}
2:  {- parameters used by the main AUPS software.                               -}
3:
4:  Procedure Fetch5210Data;
5:
6:  Var
7:      F : Text;
8:      Dummy : String;
9:
10: Begin
11:     Exec('C:\5210DATA.EXE','');
12:     If DosError <> 0 then Begin
13:         GotoXY(1,25);
14:         TextColor(Yellow);
15:         TextBackground(Red);
16:         Write('WARNING! Error in invoking RS232 support routine. ');
17:         Sound(200);
18:         Delay(200);
19:         NoSound;
20:         TextColor(Cyan);
21:         TextBackground(Black);
22:         GotoXY(1,25);
23:         Write('
24:     end;
25:     Assign(F, 'C:\5210DATA.OUT');
26:     Reset(F);
27:     If CheckforIOError then Exit;
28:     Readln(F, Data.LIAFilterTuningMode);
29:     Readln(F, Data.LIADynamicReserve);
30:     Readln(F, Dummy);
31:     Readln(F, Data.LIAFilterMode);
32:     Readln(F, Data.LIAReffreq);
33:     Readln(F, Data.LIAVernierGainStatus, Data.LIAVernierGain);
34:     Readln(F, Dummy);
35:     Readln(F, Data.LIALineFilters);
36:     Readln(F, Data.LIAOscAmplitude);
37:     Readln(F, Data.LIAOscFreq, Data.LIAOscFreqBand);
38:     Readln(F, Dummy);
39:     Readln(F, Dummy);
40:     Readln(F, Data.LIASens);
41:     Readln(F, Data.LIAFilterIC);
42:     Readln(F, Dummy);
43:     Readln(F, Data.LIAFilterRollOff);
44:     Close(F);
45: End; {- of Fetch5210Data -}
46:
47: {- End of 5210DATA.INC -}

```

## LISTING FOR FILE C:\PASCAL\INC\STRFUNCS.INC

```

1:  {- This file (STRFUNCS.INC) consists of the functions required to create  -}
2:  {- string forms of parameters needed for display in the AUPS menus.      -}
3:
4:  Function DateStr : string;
5:
6:  Var
7:      DayStr, MonthStr, YearStr : String;
8:      Year, Month, Day, DayOfWeek : Word;
9:
10: Begin
11:     GetDate(Year, Month, Day, DayOfWeek);
12:     Str(Year, YearStr);
13:     Str(Month, MonthStr);
14:     Str(Day, DayStr);
15:     If Day < 10 then DayStr := '0' + DayStr;
16:     If Month < 10 then MonthStr := '0' + MonthStr;
17:     DateStr := Concat(DayStr, '-', MonthStr, '-', YearStr);
18: End; {- of DateStr -}
19:
20: Function TimeStr : String;
21:
22: Var
23:     Hour, Min, Sec, Sec100 : Word;
24:     HourStr, MinStr : String;
25:
26: Begin
27:     GetTime(Hour, Min, Sec, Sec100);
28:     Str(Min, MinStr);
29:     If Hour > 12 then Str(Hour - 12, HourStr) else Str(Hour, HourStr);
30:     If Min < 10 then MinStr := '0' + MinStr;
31:     If Hour > 11 then MinStr := MinStr + 'pm' else MinStr := MinStr + 'am';
32:     TimeStr := HourStr + ':' + MinStr;
33: End; {- of TimeStr -}
34:
35:
36:
37: Function WavelengthStr(Wavelength : extended): string;
38:
39: Var I : string;
40:
41: Begin
42:     Str(Wavelength:1:1, I);
43:     WavelengthStr := I + ' nm';
44: End; {- of WavelengthStr -}
45:
46: Function AngleStr(Angle: extended): string;
47:
48: Var I : string;
49:
50: Begin
51:     Str(Angle:1:1, I);
52:     AngleStr := I + ' deg';
53: End; {- of AngleStr -}
54:
55: Function DelayTimeStr(DelayTime: extended): string;

```

```

56:
57:   Var I : string;
58:
59:   Begin
60:     Str(DelayTime:1:6,I);
61:     DelayTimeStr := I + ' s';
62:   End; {- of DelayTimeStr -}
63:
64:
65:   Function ElectrodeFreqStr(ElectrodeFreq: extended): string;
66:
67:   Var I : string;
68:
69:   Begin
70:     Str(ElectrodeFreq:1:2,I);
71:     ElectrodeFreqStr := I + ' Hz';
72:   End; {- of ElectrodeFreqStr -}
73:
74:   Function ElectrodeAmpStr(ElectrodeAmp: extended): string;
75:
76:   Var I : string;
77:
78:   Begin
79:     Str(ElectrodeAmp:1:2,I);
80:     ElectrodeAmpStr := I + ' U';
81:   End; {- of ElectrodeAmpStr -}
82:
83:   Function ElectrodeGapStr(ElectrodeGap: extended): string;
84:
85:   Var I : string;
86:
87:   Begin
88:     Str(ElectrodeGap:1:2,I);
89:     ElectrodeGapStr := I + ' mm';
90:   End; {- of ElectrodeGapStr -}
91:
92:   Function ViscosityStr(Viscosity: extended): string;
93:
94:   Var I : string;
95:
96:   Begin
97:     Str(Viscosity:1:3,I);
98:     ViscosityStr := I + ' cP';
99:   End; {- of ViscosityStr -}
100:
101:   Function RefIndexStr(RefIndex: extended): string;
102:
103:   Var I : string;
104:
105:   Begin
106:     Str(RefIndex:1:3,I);
107:     RefIndexStr := I;
108:   End; {- of RefIndexStr -}
109:
110:   Function TemperatureStr(Temperature: extended): string;

```



```

111:
112:   Var T : string;
113:
114:   Begin
115:     Str(Temperature:1:1,T);
116:     TemperatureStr := T + ' °C';
117:   End; {- of TemperatureStr -}
118:
119:   Function RejectionFactorStr(RejectionFactor: extended): string;
120:
121:   Var T : string;
122:
123:   Begin
124:     Str(RejectionFactor:1:1,T);
125:     RejectionFactorStr := T;
126:   End; {- of RejectionFactorStr -}
127:
128:   Function NumPointsStr(NumPoints: word): string;
129:
130:   Var T : string;
131:
132:   Begin
133:     Str(NumPoints,T);
134:     NumPointsStr := T;
135:   End; {- of NumPointsStr -}
136:
137:   Function NumSamplesStr(NumSamples: word): string;
138:
139:   Var T : string;
140:
141:   Begin
142:     Str(NumSamples,T);
143:     NumSamplesStr := T;
144:   End; {- of NumSamplesStr -}
145:
146:
147:   Function LIAReffFreqStr(Freq : LongInt): string;
148:
149:   Var T : string;
150:       Freq2 : extended;
151:
152:   Begin
153:     Freq2 := Freq / 1000;
154:     Str(Freq2:1:1,T);
155:     LIAReffFreqStr := T + ' Hz';
156:   End; {- of LIAReffFreqStr -}
157:
158:   Function LIAVernierGainStr(Status, Gain : byte) : string;
159:
160:   Var T : string;
161:
162:   Begin
163:     If Status = 1 then Str(Gain, T) else T := 'OFF';
164:     LIAVernierGainStr := T;
165:   End; {- of LIAVernierGainStr -}

```



```
166:
167: Function LIAOscAmplitudeStr(Amp : word) : string;
168:
169:   Var I : string;
170:       Amp2 : extended;
171:
172:   Begin
173:     Amp2 := Amp / 1000;
174:     Str(Amp2:1:3, I);
175:     LIAOscAmplitudeStr := I + ' U rms';
176:   End; {- of LIAOscAmplitudeStr -}
177:
178:
179:   {- End of STRFUNCS.INC -}
```

## LISTING FOR FILE C:\PASCAL\INC\MENUS24.INC

```

1:  (- This file (MENUS24.INC) contains the necessary routines for construction-)
2:  (- of the various menus required for the RUMPS software.                -)
3:
4:  Procedure MenuTitle(Title : String);
5:
6:  Begin
7:    TextColor(LightRed);
8:    GotoXY(40 - length(Title) div 2,3);
9:    Write(Title);
10: End; (- of MenuTitle -)
11:
12: Procedure MenuItem(Item : Byte; Item1Str, Item2Str : String);
13:
14: Begin
15:   TextColor(Yellow);
16:   GotoXY(30, 3 + 2 * Item);
17:   If Item = 10 then Item := 0;
18:   Write(Item, ' ');
19:   TextColor(Green);
20:   Write(Item1Str);
21:   TextColor(Red);
22:   Write(' ' + Item2Str);
23: End; (- of MenuItem -)
24:
25: Procedure UserPrompt(Prompt : String);
26:
27: Begin
28:   TextColor(Cyan);
29:   GotoXY(1,25);
30:   Write(Prompt);
31: End; (- of UserPrompt -)
32:
33: Procedure ClearUserPrompt;
34:
35: Begin
36:   TextColor(Cyan);
37:   GotoXY(1,25);
38:   Write(' ');
39: End; (- of ClearUserPrompt -)
40:
41: Procedure NotYetDefined;
42:
43: Begin
44:   UserPrompt('REQUESTED FACILITY NOT YET INSTALLED! Press SPACEBAR');
45:   ErrorBeep;
46:   Repeat Until ReadKey = ' ';
47: End; (- of NotYetDefined -)
48:
49: Procedure DisplayMainTitle;
50:
51: Begin
52:   TextColor(LightGray);
53:   GotoXY(1,1);
54:   Write('Phase Analysis Light Scattering (PALS) Controller      (Version 2.4 Mar 1990)');
55: End; (- of DisplayMainTitle -)

```

```

56:
57: Procedure DisplayMainMenu;
58:
59: Begin
60:   ClrScr;
61:   DisplayMainTitle;
62:   MenuTitle('MAIN MENU');
63:   MenuItem(1,'Parameters/Settings','');
64:   MenuItem(2,'Data Storage/Retrieval','');
65:   MenuItem(3,'View LIA Signals','');
66:   MenuItem(4,'Data Processing','');
67:   MenuItem(5,'Run Experiment','');
68:   MenuItem(6,'Multi Run ','');
69:   MenuItem(7,'Quit','');
70:   MenuItem(8,'Shell to DOS','');
71:   ClearUserPrompt;
72:   UserPrompt('Make Selection (1-8)');
73: End; {- of DisplayMainMenu -}
74:
75: Procedure DisplayParameterMenu;
76:
77: Begin
78:   ClrScr;
79:   DisplayMainTitle;
80:   MenuTitle('PARAMETER MENU');
81:   MenuItem(1,'Optical','');
82:   MenuItem(2,'Signal Processing','');
83:   MenuItem(3,'Electrode Drive','');
84:   MenuItem(4,'Sample/Other','');
85:   MenuItem(5,'Write to Disc','');
86:   MenuItem(6,'Read from Disc','');
87:   MenuItem(7,'Main Menu','');
88:   ClearUserPrompt;
89:   UserPrompt('Make Selection (1-7)');
90: End; {- of DisplayParameterMenu -}
91:
92: Procedure DisplayDataStorageMenu;
93:
94: Begin
95:   ClrScr;
96:   DisplayMainTitle;
97:   MenuTitle('DATA STORAGE MENU');
98:   MenuItem(1,'Retrieve Data','');
99:   MenuItem(2,'Store Data','');
100:  MenuItem(3,'Create ''Grapher'' file','');
101:  MenuItem(4,'Change Directory','');
102:  MenuItem(5,'Main Menu','');
103:  ClearUserPrompt;
104:  UserPrompt('Make Selection (1-5)');
105: End; {- of DisplayDataStorageMenu -}
106:
107: Procedure DisplayDataProcessingMenu;
108:
109: Begin
110:   ClrScr;

```

```

111:   DisplayMainTitle;
112:   MenuTitle('DATA PROCESSING MENU');
113:   MenuItem(1,'Fit AUPD etc.','');
114:   MenuItem(2,'Display Amplitude Histogram','');
115:   MenuItem(3,'Display Real Comp. Correlogram (Normal)','');
116:   MenuItem(4,'Display Real Comp. Correlogram (Semilog)','');
117:   MenuItem(5,'Main menu','');
118:   ClearUserPrompt;
119:   UserPrompt('Make Selection (1-5)');
120: End; {- of DisplayDataProcessingMenu -}
121:
122: Procedure DisplayOpticalMenu;
123:
124: Begin
125:   ClrScr;
126:   DisplayMainTitle;
127:   MenuTitle('OPTICAL PARAMETERS');
128:   MenuItem(1,'Laser wavelength:',WavelengthStr(Data.Wavelength));
129:   MenuItem(2,'Scattering angle:',AngleStr(Data.Angle));
130:   MenuItem(3,'Lens aperature:',LensApertureStr[Data.LensAperture]);
131:   MenuItem(4,'Pinhole size:',PinHoleSizeStr[Data.PinHoleSize]);
132:   MenuItem(5,'Lens tubes:',LensTubeStr[Data.LensTubes]);
133:   MenuItem(6,'Sample cell:',Data.CellType);
134:   MenuItem(7,'Previous menu','');
135:   ClearUserPrompt;
136:   UserPrompt('Make Selection (1-7)');
137: End; {- of DisplayOpticalMenu -}
138:
139: Procedure DisplaySignalProcessingMenu;
140:
141: Begin
142:   ClrScr;
143:   DisplayMainTitle;
144:   MenuTitle('SIGNAL PROCESSING PARAMETERS');
145:   MenuItem(1,'Filter tuning mode:',LIAFilterTuningModeStr[Data.LIAFilterTuningMode]);
146:   MenuItem(2,'Dynamic reserve:',LIADynamicReserveStr[Data.LIADynamicReserve]);
147:   MenuItem(3,'Filter mode:',LIAFilterModeStr[Data.LIAFilterMode]);
148:   MenuItem(4,'Reference frequency:',LIAReffreqStr[Data.LIAReffreq]);
149:   MenuItem(5,'Vernier gain:',LIAVernierGainStr[Data.LIAVernierGainStatus,Data.LIAVernierGain]);
150:   MenuItem(6,'Line filters:',LIALineFiltersStr[Data.LIALineFilters]);
151:   MenuItem(7,'Oscillator amplitude:',LIAOscAmplitudeStr[Data.LIAOscAmplitude]);
152:   MenuItem(8,'Input sensitivity:',LIASensStr[Data.LIASens]);
153:   MenuItem(9,'filter time constant:',LIAFilterICStr[Data.LIAFilterIC]);
154:   MenuItem(10,'filter rolloff rate:',LIAFilterRolloffStr[Data.LIAFilterRolloff]);
155:   ClearUserPrompt;
156:   UserPrompt('Press SPACEBAR to continue');
157: End; {- of DisplaySignalProcessingMenu -}
158:
159: Procedure DisplayElectrodeDriveMenu;
160:
161: Begin
162:   ClrScr;
163:   DisplayMainTitle;
164:   MenuTitle('ELECTRODE DRIVE PARAMETERS');
165:   MenuItem(1,'Electrode frequency:',ElectrodeFreqStr[Data.ElectrodeFreq]);

```

```

166:   MenuItem(2,'Electrode amplitude:',ElectrodeAmpStr(Data.ElectrodeAmp));
167:   MenuItem(3,'Electrode spacing:',ElectrodeGapStr(Data.ElectrodeGap));
168:   MenuItem(4,'Electrode type:',Data.ElectrodeType);
169:   MenuItem(5,'Previous menu','');
170:   ClearUserPrompt;
171:   UserPrompt('Make Selection (1-5)');
172: End; {- of DisplayElectrodeDriveMenu -}
173:
174: Procedure DisplayOtherMenu;
175:
176: Begin
177:   ClrScr;
178:   DisplayMainTitle;
179:   MenuTitle('SAMPLE/OTHER PARAMETERS');
180:   MenuItem(1,'Sample Name:',Data.SampleName);
181:   MenuItem(2,'Sample ref. index:',RefIndexStr(Data.RefIndex));
182:   MenuItem(3,'Sample temperature:',TemperatureStr(Data.Temperature));
183:   MenuItem(4,'Sample viscosity:',ViscosityStr(Data.Viscosity));
184:   MenuItem(5,'Measurement date:',Data.Date);
185:   MenuItem(6,'Measurement time:',Data.Time);
186:   MenuItem(7,'Number of channels:',NumPointsStr(Data.NumPoints));
187:   MenuItem(8,'Number of samples:',NumSamplesStr(Data.NumSamples));
188:   MenuItem(9,'Delay Time:',DelayTimeStr(Data.DelayTime));
189:   MenuItem(10,'Previous menu','');
190:   ClearUserPrompt;
191:   UserPrompt('Make Selection (1-10)');
192: End; {- of DisplayOtherMenu -}
193:
194: {- End of MENUS.INC -}

```



## LISTING FOR FILE C:\PASCAL\INC\INPUTS24.INC

```

1:  {- This file (INPUTS.INC) defines the procedures which allow the user to  -}
2:  {- enter new parameters etc. In each case, function key f1 is redefined as -}
3:  {- the previous parameter value. Inputs are checked for valid range, form -}
4:  {- etc. Invalid responses result in a repeated input request.             -}
5:
6:  Procedure InputWavelength;
7:
8:  Begin
9:      Repeat
10:         Begin
11:             DisplayOpticalMenu;
12:             ClearUserPrompt;
13:             UserPrompt('Enter laser wavelength (nm): ');
14:             DefFuncKeyExtended(Data.Wavelength,1,1);
15:             Readln(Data.Wavelength);
16:         End;
17:         Until IOResult = 0;
18:     End; {- of InputWavelength -}
19:
20: Procedure InputDelayTime;
21:
22: Begin
23:     Repeat
24:         Begin
25:             DisplayOtherMenu;
26:             ClearUserPrompt;
27:             UserPrompt('Enter delay time (s): ');
28:             DefFuncKeyExtended(Data.DelayTime,1,6);
29:             Readln(Data.DelayTime);
30:         End;
31:         Until IOResult = 0;
32:     End; {- of InputDelayTime -}
33:
34: Procedure InputElectrodefreq;
35:
36: Begin
37:     Repeat
38:         Begin
39:             DisplayElectrodeDriveMenu;
40:             ClearUserPrompt;
41:             UserPrompt('Enter electrode frequency (Hz): ');
42:             DefFuncKeyExtended(Data.Electrodefreq,1,2);
43:             Readln(Data.Electrodefreq);
44:         End;
45:         Until IOResult = 0;
46:     End; {- of InputElectrodefreq -}
47:
48: Procedure InputElectrodeAmp;
49:
50: Var XIOResponse : byte;
51:     Voltage, MaxAmp, MinAmp : extended;
52:     loop : word;
53:
54: Begin
55:     DisplayElectrodeDriveMenu;

```



```

56:   ClearUserPrompt;
57:   Ext_Irig := False;
58:   Set_DIO(0, Digital_In);
59:   Read_DIO(0, XIOResponse);
60:   Ext_Irig := True;
61:   UserPrompt('Determining electrode drive amplitude. Please wait. ');
62:   MaxAmp := 0; MinAmp := 0;
63:   For loop := 1 to 1000 do begin
64:     AD_Value(3, Voltage);
65:     If Voltage > MaxAmp then MaxAmp := Voltage;
66:     If Voltage < MinAmp then MinAmp := Voltage;
67:   End;
68:   Data.ElectrodeAmp := (MaxAmp - MinAmp)/2;
69:   If (XIOResponse and 1) = 0 then Data.ElectrodeAmp := Data.ElectrodeAmp / 0.09436
70:     else Data.ElectrodeAmp := Data.ElectrodeAmp / 0.931;
71:   If Data.ElectrodeAmp < 0.05 then Data.ElectrodeAmp := 0;
72: End; {- of InputElectrodeAmp -}
73:
74: Procedure InputElectrodeGap;
75:
76: Begin
77:   Repeat
78:     Begin
79:       DisplayElectrodeDriveMenu;
80:       ClearUserPrompt;
81:       UserPrompt('Enter electrode gap (mm): ');
82:       DefFuncKeyExtended(Data.ElectrodeGap,1,2);
83:       Readln(Data.ElectrodeGap);
84:     End;
85:     Until IOResult = 0;
86: End; {- of InputElectrodeGap -}
87:
88: Procedure InputElectrodeType;
89:
90: Begin
91:   Repeat
92:     Begin
93:       DisplayElectrodeDriveMenu;
94:       ClearUserPrompt;
95:       UserPrompt('Enter electrode type (Max 20 chars): ');
96:       DefFuncKey(Data.ElectrodeType);
97:       Readln(Data.ElectrodeType);
98:     End;
99:     Until IOResult = 0;
100: End; {- of InputElectrodeType -}
101:
102: Procedure InputSampleName;
103:
104: Begin
105:   Repeat
106:     Begin
107:       DisplayOtherMenu;
108:       ClearUserPrompt;
109:       UserPrompt('Enter sample name (Max 30 chars): ');
110:       DefFuncKey(Data.SampleName);

```

```

111:      Readln(Data.SampleName);
112:      End;
113:      Until IOResult = 0;
114:      End; {- of InputSampleName -}
115:
116:      Procedure InputCellType;
117:
118:      Begin
119:          Repeat
120:              Begin
121:                  DisplayOpticalMenu;
122:                  ClearUserPrompt;
123:                  UserPrompt('Enter sample cell type (Max 20 chars): ');
124:                  DefFuncKey(Data.CellType);
125:                  Readln(Data.CellType);
126:              End;
127:              Until IOResult = 0;
128:          End; {- of InputCellType -}
129:
130:      Procedure InputViscosity;
131:
132:      Begin
133:          Repeat
134:              Begin
135:                  DisplayOtherMenu;
136:                  ClearUserPrompt;
137:                  UserPrompt('Enter sample viscosity (cP): ');
138:                  DefFuncKeyExtended(Data.Viscosity,1,3);
139:                  Readln(Data.Viscosity);
140:              End;
141:              Until IOResult = 0;
142:          End; {- of InputViscosity -}
143:
144:      Procedure InputRefIndex;
145:
146:      Begin
147:          Repeat
148:              Begin
149:                  DisplayOtherMenu;
150:                  ClearUserPrompt;
151:                  UserPrompt('Enter sample ref. index: ');
152:                  DefFuncKeyExtended(Data.RefIndex,1,3);
153:                  Readln(Data.RefIndex);
154:              End;
155:              Until IOResult = 0;
156:          End; {- of InputRefIndex -}
157:
158:      Procedure InputTemperature;
159:
160:      Begin
161:          Repeat
162:              Begin
163:                  DisplayOtherMenu;
164:                  ClearUserPrompt;
165:                  UserPrompt('Enter sample temperature (°C): ');

```

```

166:      DefFuncKeyExtended(Data.Temperature,1,1);
167:      Readln(Data.Temperature);
168:      End;
169:      Until IOResult = 0;
170: End; {- of InputTemperature -}
171:
172: Procedure InputDate;
173:
174: Begin
175:   Repeat
176:     Begin
177:       DisplayOtherMenu;
178:       ClearUserPrompt;
179:       UserPrompt('Enter sample date: ');
180:       DefFuncKey(Data.Date);
181:       Readln(Data.Date);
182:     End;
183:     Until IOResult = 0;
184: End; {- of InputDate -}
185:
186: Procedure InputTime;
187:
188: Begin
189:   Repeat
190:     Begin
191:       DisplayOtherMenu;
192:       ClearUserPrompt;
193:       UserPrompt('Enter sample time: ');
194:       DefFuncKey(Data.Time);
195:       Readln(Data.Time);
196:     End;
197:     Until IOResult = 0;
198: End; {- of InputTime -}
199:
200: Procedure InputNumPoints;
201:
202: Begin
203:   Repeat
204:     Begin
205:       DisplayOtherMenu;
206:       ClearUserPrompt;
207:       UserPrompt('Enter number of channels in AUPD/AUPS functions: ');
208:       DefFuncKeyWord(Data.NumPoints);
209:       Readln(Data.NumPoints);
210:     End;
211:     Until (IOResult = 0) and (Data.NumPoints > 16) and (Data.NumPoints < 251);
212: End; {- of InputNumPoints -}
213:
214: Procedure InputNumSamples;
215:
216: Begin
217:   Repeat
218:     Begin
219:       DisplayOtherMenu;
220:       ClearUserPrompt;

```

```

221:     UserPrompt('Enter number of samples: ');
222:     DefFuncKeyWord(Data.NumSamples);
223:     Readln(Data.NumSamples);
224:     End;
225:     Until (IOResult = 0) and (Data.NumSamples > 0);
226: End; {- of InputNumSamples -}
227:
228: Procedure InputRejectionFactor;
229:
230: Begin
231:     Repeat
232:         Begin
233:             DisplayOtherMenu;
234:             ClearUserPrompt;
235:             UserPrompt('Enter rejection factor (0 to 9.999): ');
236:             DefFuncKeyExtended(Data.RejectionFactor,1,3);
237:             Readln(Data.RejectionFactor);
238:         End;
239:         Until (IOResult = 0) and (Data.RejectionFactor >= 0)
240:             and (Data.RejectionFactor < 10);
241:     End; {- of InputRejectionFactor -}
242:
243:
244: Procedure InputAngle;
245:
246:
247: Begin
248:     Repeat
249:         Begin
250:             DisplayOpticalMenu;
251:             ClearUserPrompt;
252:             UserPrompt('Enter scattering angle in air (deg): ');
253:             DefFuncKeyExtended(Data.Angle,1,2);
254:             Readln(Data.Angle);
255:         End;
256:         Until IOResult = 0;
257:     End; {- of InputAngle -}
258:
259: Procedure InputLensAperture;
260:
261: Var loop, temp : byte;
262:
263: Begin
264:     Repeat
265:         Begin
266:             ClrScr;
267:             DisplayMainTitle;
268:             MenuTitle('LENS APERTURE OPTIONS');
269:             for loop := 1 to 9 do MenuItem(loop, LensApertureStr[loop],'');
270:             ClearUserPrompt;
271:             UserPrompt('Select aperture (1-9):');
272:             Readln(temp);
273:             if temp = 0 then temp := 10;
274:         End;
275:         Until (IOResult = 0) and (temp > 0) and (temp<12);

```

```

276:     Data.LensAperture := temp;
277: End; {- of InputLensAperture -}
278:
279: Procedure InputPinHoleSize;
280:
281:   Var loop, temp : byte;
282:
283:   Begin
284:     Repeat
285:       Begin
286:         ClrScr;
287:         DisplayMainTitle;
288:         MenuTitle('PINHOLE SIZE OPTIONS');
289:         for loop := 1 to 3 do MenuItem(loop, PinHoleSizeStr[loop], '');
290:         ClearUserPrompt;
291:         UserPrompt('Select pinhole size (1-3):');
292:         Readln(temp);
293:       End;
294:       Until (IOResult = 0) and (temp > 0) and (temp < 4);
295:       Data.PinHoleSize := temp;
296:     End; {- of InputPinHoleSize -}
297:
298: Procedure InputLensTubes;
299:
300:   Var loop, temp : byte;
301:
302:   Begin
303:     Repeat
304:       Begin
305:         ClrScr;
306:         DisplayMainTitle;
307:         MenuTitle('LENS TUBES OPTIONS');
308:         for loop := 1 to 8 do MenuItem(loop, LensTubeStr[loop], '');
309:         ClearUserPrompt;
310:         UserPrompt('Select lens tubes (1-8):');
311:         Readln(temp);
312:       End;
313:       Until (IOResult = 0) and (temp > 0) and (temp < 9);
314:       Data.LensTubes := temp;
315:     End; {- of InputLensTubes -}
316:
317:
318:
319: {- End of INPUTS.INC -}

```



## LISTING FOR FILE C:\PASCAL\INC\ENTMENUS.INC

```

1:  {- This file (ENTMENUS.INC) contains the procedures defining the control  -}
2:  {- loops for the various RWPS sub-menus.                                -}
3:
4:  Procedure EnterOpticalMenu;
5:
6:  Begin
7:      Repeat
8:          Begin
9:              DisplayOpticalMenu;
10:             Case ReadKey of
11:                 '1' : InputWavelength;
12:                 '2' : InputAngle;
13:                 '3' : InputLensAperture;
14:                 '4' : InputPinHoleSize;
15:                 '5' : InputLensTubes;
16:                 '6' : InputCellType;
17:                 '7' : Exit;
18:             else
19:                 ErrorBeep;
20:             end;
21:         end;
22:     Until False;
23: End; {- of EnterOpticalMenu -}
24:
25: Procedure EnterSignalProcessingMenu;
26:
27: Begin
28:     Repeat
29:         Begin
30:             DisplaySignalProcessingMenu;
31:             Case ReadKey of
32:                 ' ' : Exit;
33:             else
34:                 ErrorBeep;
35:             end;
36:         end;
37:     Until False;
38: End; {- of EnterSignalProcessingMenu -}
39:
40: Procedure EnterElectrodeDriveMenu;
41:
42: Begin
43:     Repeat
44:         Begin
45:             DisplayElectrodeDriveMenu;
46:             Case ReadKey of
47:                 '1' : InputElectrodeFreq;
48:                 '2' : InputElectrodeAmp;
49:                 '3' : InputElectrodeGap;
50:                 '4' : InputElectrodeType;
51:                 '5' : Exit;
52:             else
53:                 ErrorBeep;
54:             end;
55:         end;

```

```
56:      Until False;
57: End; {- of EnterElectrodeDriveMenu -}
58:
59: Procedure EnterOtherMenu;
60:
61: Begin
62:   Repeat
63:     Begin
64:       DisplayOtherMenu;
65:       Case ReadKey of
66:         '1' : InputSampleName;
67:         '2' : InputRefIndex;
68:         '3' : InputTemperature;
69:         '4' : InputViscosity;
70:         '5' : InputDate;
71:         '6' : InputTime;
72:         '7' : InputNumPoints;
73:         '8' : InputNumSamples;
74:         '9' : InputDelayTime;
75:         '0' : Exit;
76:       else
77:         ErrorBeep;
78:       end;
79:     end;
80:   Until False;
81: End; {- of EnterOtherMenu -}
82:
83: {- End of ENTMENU.INC -}
```

## LISTING FOR FILE C:\PASCAL\INC\FITAWPS2.INC

```

1:  (- This file (FITAWPS2.INC) contains the procedures/functions for the      -)
2:  (- implementation of the data processing section of the AWPS software.    -)
3:
4:  Procedure SmallBox(X, Y : integer);
5:
6:  Begin
7:      Rectangle(X-1, Y-1, X+1, Y+1);
8:  End; (- of SmallBox -)
9:
10: Procedure DisplayHistogram(Histogram : DerivArray);
11:
12: Var
13:     loop : word;
14:     YMax : extended;
15:
16: Begin
17:     SetGraphMode(CGRAHi);
18:     ClearDevice;
19:     YMax := 0;
20:     for loop := 0 to 99 do if Histogram[loop] > YMax then YMax := Histogram[loop];
21:     Rectangle(220, 50, 420, 150);
22:     for loop := 0 to 99 do
23:         begin
24:             if loop = 0 then MoveTo(220,150)
25:             else
26:                 LineTo(220 + loop * 2, 150 - Trunc(100 * Histogram[loop] / YMax));
27:         end;
28:     SetTextJustify(CenterText, TopText);
29:     OutTextXY(320, 0, 'Amplitude Histogram');
30:     OutTextXY(320, 180, 'Press SPACEBAR to continue');
31:     Repeat Until ReadKey = ' ';
32:     TextMode(3);
33: End; (- of DisplayHistogram -)
34:
35: Procedure DisplayCorrelogram(AmpCorrData : DerivArray; SemiLog : boolean);
36:
37: Var
38:     loop : word;
39:     YMin, YMax : extended;
40:
41: Begin
42:     SetGraphMode(CGRAHi);
43:     ClearDevice;
44:     (- Take natural log of data if Semilog = True -)
45:     If SemiLog then for loop := 0 to Data.NumPoints - 1 do
46:         If AmpCorrData[loop] <= 0 then AmpCorrData[loop] := 0
47:         else AmpCorrData[loop] := Ln(AmpCorrData[loop]);
48:     YMax := AmpCorrData[0];
49:     YMin := AmpCorrData[0];
50:     for loop := 0 to Data.NumPoints - 1 do if AmpCorrData[loop] > YMax then YMax := AmpCorrData[loop];
51:     for loop := 0 to Data.NumPoints - 1 do if AmpCorrData[loop] < YMin then YMin := AmpCorrData[loop];
52:     Rectangle(320 - (Data.NumPoints * 2), 50, 320 + (Data.NumPoints * 2), 150);
53:     for loop := 0 to Data.NumPoints - 1 do
54:         begin
55:             if loop = 0 then MoveTo(320 - Data.NumPoints * 2, 50)

```

```

56:         else if (Not(Semilog) or (AmpCorrData[loop] < 0)) then
57:             SmallBox(320 - Data.NumPoints * 2 + loop * 4,
58:                 150 - Trunc(100 * (AmpCorrData[loop] - YMin) / (YMax - YMin)));
59:         end;
60:         SetTextJustify(CenterText, TopText);
61:         If Not(Semilog) then OutTextXY(320, 0, 'Real Component Correlogram (Linear Plot)')
62:             else OutTextXY(320, 0, 'Real Component Correlogram (Semilog Plot)');
63:         OutTextXY(320, 180, 'Press SPACEBAR to continue');
64:         Repeat Until ReadKey = ' ';
65:         TextMode(3);
66:     End; {- of DisplayCorrelogram -}
67:
68: Procedure DisplaySwinbournePlot(Title : String; X, Y : INColumnVector; N : word);
69:
70: Var
71:     loop : word;
72:     XMax, XMin, YMax, YMin : extended;
73:
74: Begin
75:     SetGraphMode(CG64Hi);
76:     ClearDevice;
77:     YMax := Y[1]; YMin := Y[1];
78:     XMax := X[1]; XMin := X[1];
79:     for loop := 1 to N do begin
80:         If X[loop] > XMax then XMax := X[loop];
81:         If X[loop] < XMin then XMin := X[loop];
82:         If Y[loop] > YMax then YMax := Y[loop];
83:         If Y[loop] < YMin then YMin := Y[loop];
84:     end;
85:     Rectangle(220, 50, 420, 150);
86:     for loop := 1 to N do
87:         SmallBox(220 + Trunc(200 * (X[loop] - XMin) / (XMax - XMin)),
88:             150 - Trunc(100 * (Y[loop] - YMin) / (YMax - YMin)));
89:         SetTextJustify(CenterText, TopText);
90:         OutTextXY(320, 0, Title);
91:         OutTextXY(320, 180, 'Press SPACEBAR to continue');
92:     End; {- of DisplaySwinbournePlot -}
93:
94: Procedure EnterFitAWPSMenu(QPrompts : boolean);
95:
96: Var
97:     AWPSFit : DerivArray;
98:     A, B, B1, B2, C, D, SD, SN, Variance, Q, Const1, Const2, Const3,
99:     Velocity, Diameter, Mobility : extended;
100:     Err : Byte;
101:
102:
103: Procedure Display_Fit(Var AWPS, AWPSFit : DerivArray;
104:     NumPoints : Word);
105:
106: Var
107:     loop : word;
108:     YMin, YMax : extended;
109:
110: Begin

```

```

111: SetGraphMode(CGAI);
112: ClearDevice;
113: YMax := 0;
114: for loop := 0 to NumPoints - 1 do begin
115:   if AvAUPS[loop] < YMin then YMin := AvAUPS[loop];
116:   if AvAUPS[loop] > YMax then YMax := AvAUPS[loop];
117: end;
118: Rectangle(320 - NumPoints, 50, 320 + NumPoints, 150);
119: for loop := 0 to NumPoints - 1 do
120:   SmallBox(320 - NumPoints + loop * 2,
121:     150 - Trunc(100 * (AvAUPS[loop] - YMin) / (YMax - YMin)));
122: for loop := 0 to NumPoints - 1 do
123:   if loop = 0 then MoveTo(320 - NumPoints, 150)
124:   else LineTo(320 - NumPoints + loop * 2,
125:     150 - Trunc(100 * (AUPSFit[loop] - YMin)
126:       / (YMax - YMin)));
127: End; {- of Display_fit -}
128:
129:
130: Procedure Fit_AUPS(Var AmpCorr, AUPD, AUPS, TimeData : DerivArray;
131:   NumPoints : Word;
132:   we, lag : extended;
133:   Var AUPSFit : DerivArray;
134:   Var A, B1, B2, C, D, SN : Extended;
135:   Var SD : Extended;
136:   Var Variance : Extended;
137:   Var Err : byte;
138:   fit : byte);
139:
140: {- This procedure uses the non-linear least squares unit to obtain the 3 -}
141: {- fit parameters, A, B, C, and D for the AUPS equation... -}
142:
143: Var
144:   LSQXData, LSQYData, LSQYFit, LSQResiduals : TNColumnVector;
145:   SwinXData, SwinYData : TNColumnVector;
146:   LSQSolution : TNRowVector;
147:   CorrectedAmpCorr : DerivArray;
148:   MaxDiffData, MinDiffData : extended;
149:   LSQpts, NextX, Loop, Loop2, Loop3 : word;
150:
151: Procedure SwinbournePlot( YData : DerivArray;
152:   Var SwinXData, SwinYData : TNColumnVector;
153:   Var NumOfSwinPoints : word);
154:
155: Var
156:   loop : word;
157:
158: Begin
159:   NumOfSwinPoints := 0;
160:   While (YData[NumOfSwinPoints] > 0) and (NumOfSwinPoints < NumPoints) do
161:     Inc(NumOfSwinPoints);
162:   Dec(NumOfSwinPoints);
163:   If NumOfSwinPoints = 0 then exit;
164:   For loop := 1 to NumOfSwinPoints do begin
165:     SwinXData[loop] := YData[loop - 1];

```



```

166:     SwinYData[loop] := YData[loop];
167:     end;
168: End; {- of SwinbournePlot -}
169:
170:
171: Begin
172:     {- Determine fit to amplitude correlation function -}
173:     {- Set up Swinbourne data arrays -}
174:     SwinbournePlot(AmpCorr, SwinXData, SwinYData, LSQpts);
175:     LSQpts := LSQpts div 2; {- Use half correlation data since latter half -}
176:                             {- tends to be rather noisy. -}
177:     If LSQpts <> 0 then begin
178:         for loop := 1 to LSQpts do begin
179:             LSQXData[loop] := TimeData[loop - 1];
180:             LSQYData[loop] := ln(AmpCorr[loop - 1]);
181:         end;
182:         LeastSquares(LSQpts, LSQXData, LSQYData,
183:                     0, 0, 2, LSQSolution, LSQYFit, LSQResiduals, SD,
184:                     Variance, Err, 3);
185:         GlobalIntercept := 2 * LSQSolution[3] / Sqr(LSQSolution[2]);
186:         GlobalGradient := - LSQSolution[2] / exp(LSQSolution[1]);
187:         MenuTitle('REAL COMP. CORRELATION DATA');
188:         GotoXY(1,6);
189:         TextColor(DarkGray);
190:         Writeln('          / s      C( )      ln(C( ))');
191:         for loop := 1 to 10 do Writeln('          ',LSQXData[loop]:11,' ',
192:             AmpCorr[loop-1]:1:5,' ',LSQYData[loop]:1:5);
193:         Writeln('');
194:         Writeln('          Least squares fit: f( ) = ',LSQSolution[2]:1:5,' + ',LSQSolution[1]:1:5);
195:         Writeln('          Data points used in fit = ',LSQpts,' ');
196:         Writeln('          Dq2 = ',GlobalGradient:1:1);
197:         Writeln('          "Polydispersity" factor = ',GlobalIntercept:1:3);
198:         ClearUserPrompt;
199:         UserPrompt('Press SPACEBAR to continue. ');
200:         If QPrompts then Repeat Until ReadKey = ' ';
201:     end;
202:     for loop := 1 to NumPoints do begin
203:         LSQXData[loop] := TimeData[loop - 1];
204:         LSQYData[loop] := AWPSE[loop - 1];
205:     end;
206:     DisplaySwinbournePlot('Experimental AWPSE function', LSQXData, LSQYData, NumPoints);
207:     If QPrompts then Repeat Until ReadKey = ' ';
208:     TextMode(3);
209:     For loop := 1 to NumPoints do begin
210:         LSQXData[loop] := TimeData[loop - 1];
211:         LSQYData[loop] := AWPSE[loop - 1];
212:     end;
213:     If Data.ElectrodeAmp = 0 then
214:         LeastSquares(NumPoints, LSQXData, LSQYData, we, lag, 2, LSQSolution, LSQYFit,
215:                     LSQResiduals, SD, Variance, Err, Fit)
216:     else
217:         LeastSquares(NumPoints, LSQXData, LSQYData, we, lag, 4, LSQSolution, LSQYFit,
218:                     LSQResiduals, SD, Variance, Err, Fit);
219:     For loop := 1 to NumPoints do AWPSEFit[loop - 1] := LSQYFit[loop];
220:     A := LSQSolution[1];

```

```

221:   If Data.ElectrodeAmp <> 0 then begin
222:       B1 := LSQSolution[3];
223:       B2 := LSQSolution[4];
224:   end
225:   else begin
226:       B1 := 0;
227:       B2 := 0;
228:   end;
229:   C := LSQSolution[2];
230:   Display_Fit(Data.AWPDData, AWPSFit, Data.NumPoints);
231:   SetTextJustify(CenterText, TopText);
232:   OutTextXY(320,0,'Least squares fit and experimental data');
233:   OutTextXY(320,8,'(AWPD function)');
234:   OutTextXY(320,180,'Press SPACEBAR to continue');
235:   {- Now calculate Dq2 via AWPS -}
236:   {- Determine intercept value for S/N ratio -}
237:   SN := 1 / Sqrt(Abs(AWPS[1]));
238:   for loop := 1 to NumPoints - 1 do begin
239:       LSQXData[loop] := TimeData[loop];
240:       LSQYData[loop] := AWPS[loop] - AWPS[1]; {- Remove noise component -}
241:       LSQYData[loop] := LSQYData[loop] - Sqr(A * TimeData[loop]) * pi / 4;
242:   end;
243:   LeastSquares(NumPoints - 1, LSQXData, LSQYData, 0, 0, 2, LSQSolution, LSQYFit,
244:               LSQResiduals, SD, Variance, Err, 3);
245:   D := LSQSolution[2];
246:   for loop := 0 to NumPoints - 1 do begin
247:       LSQXData[loop + 1] := TimeData[loop];
248:       LSQYData[loop + 1] := AWPS[loop];
249:   end;
250:   If QPrompts then Repeat Until ReadKey = ' ';
251:   TextMode(3);
252: End; {- of Fit_AWPS -}
253:
254: Begin
255:   ClrScr;
256:   DisplayMainTitle;
257:   UserPrompt('Fitting data using 4-parameter least squares...');
258:   Fit_AWPS(Data.AmpCorr, Data.AWPDData, Data.AWPSData, Data.TimeData,
259:           Data.NumPoints, Data.ElectrodeFreq * 2 * pi, 0, AWPSFit, A, B1, B2,
260:           C, D, SN, SD, Variance, Err, 1);
261:   B := Sqrt(B1*B1 + B2*B2);
262:   {- Determine if mobility is negative -}
263:   If B1 > 0 then B := - B;
264:   DisplayMainTitle;
265:   MenuTitle('LEAST SQUARES FIT');
266:   Q := 4 * pi * Sin(Data.Angle * pi / 180) / (Data.Wavelength
267:       * 1E-09);
268:   Const1 := Q;
269:   Const2 := Q * Data.ElectrodeAmp / (Data.ElectrodeGap * 1E-3)
270:       / (Data.ElectrodeFreq * 2 * pi);
271:   Const3 := 1.38e-23 * (Data.Temperature + 273.15) / (3e-3 * pi
272:       * Data.Viscosity) * Q * Q;
273:   GlobalDiameter := 0;
274:   If D <> 0 then GlobalDiameter := Const3 / D / 1E-9;
275:   If GlobalGradient <> 0 then GlobalGradient := Const3 / GlobalGradient / 1E-9;

```

```

276:   If Data.ElectrodeAmp = 0 then GlobalMobility := 0 else
277:     GlobalMobility := B / Const2;
278:   GlobalVelocity := (A / Const1) / 1e-6;
279:   GlobalSD := SD;
280:   GlobalSN := SN;
281:   TextColor(DarkGray);
282:   GotoXY(1,6);
283:   Writeln('Fit parameters:');
284:   Writeln('      AJPD      A: ',A:1:5);
285:   Writeln('      "      B1: ',B1:1:5);
286:   Writeln('      "      B2: ',B2:1:5);
287:   Writeln('      "      C: ',C:1:5);
288:   Writeln('Amp. corr. diam/nm: ',GlobalGradient:6:1);
289:   Writeln('      " polydisp: ',GlobalIntercept:1:5);
290:   Writeln('');
291:   Write('Pseudo lag value : ');
292:   If Data.ElectrodeAmp < 0 then Writeln((ArcTan(b2/b1)*180/pi):1:2) else
293:     Writeln('UNDEFINED');
294:   Writeln('Signal/noise ratio: ',SN:1:3);
295:   Writeln('');
296:   Writeln('Velocity ( m/s)   : ',GlobalVelocity:1:2);
297:   Writeln('Diameter (nm)      : ',GlobalDiameter:1:1);
298:   Writeln('Mobility (m2/Us)   : ',GlobalMobility:11);
299:   Writeln('');
300:   Writeln('');
301:   Writeln('Standard Deviation: ',SD:11);
302:   Writeln('Variance           : ',Variance:11);
303:   Writeln('Err                : ',Err);
304:   GotoXY(40,6);
305:   Write('Sample: ',Data.SampleName);
306:   GotoXY(40,7);
307:   Write('Date: ',Data.Date);
308:   GotoXY(40,8);
309:   Write('Time: ',Data.Time);
310:   GotoXY(40,9);
311:   Write('Temperature: ',TemperatureStr(Data.Temperature));
312:   GotoXY(40,10);
313:   Write('Viscosity: ',ViscosityStr(Data.Viscosity));
314:   GotoXY(40,11);
315:   Write('Ref. index: ',RefIndexStr(Data.RefIndex));
316:   GotoXY(40,12);
317:   Write('Electrode freq: ',ElectrodeFreqStr(Data.ElectrodeFreq));
318:   GotoXY(40,13);
319:   Write('Electrode amplitude: ',ElectrodeAmpStr(Data.ElectrodeAmp));
320:   GotoXY(40,14);
321:   Write('Electrode spacing: ',ElectrodeGapStr(Data.ElectrodeGap));
322:   GotoXY(40,15);
323:   Write('Delay time: ',DelayTimeStr(Data.DelayTime));
324:   GotoXY(40,16);
325:   Write('No. of channels: ',NumPointsStr(Data.NumPoints));
326:   GotoXY(40,17);
327:   Write('No. of samples: ',NumSamplesStr(Data.NumSamples));
328:   GotoXY(40,19);
329:   Write('Stored at ',CurrentDirectory,'\\',Data.FileName);
330:   GotoXY(40,20);

```

```
331:   Write('      on ',CurrentVolume);
332:   UserPrompt('Press SPACEBAR to return to menu');
333:   If QPrompts then Repeat Until ReadKey = ' ' else
334:     If AutoTrack then
335:       begin
336:         CentreOffsetInc := A / (2 * pi * FreqToVoltsfactor);
337:         CentreOffset := CentreOffset + CentreOffsetInc;
338:         DA_Value(0, CentreOffset);
339:       end;
340:   End; (- of EnterFitAUPSMenus -)
341:
342: Procedure EnterDisplayFitMenu;
343:
344:   Begin
345:     NotYetDefined;
346:   End; (- of EnterDisplayFitMenu -)
347:
348:   (- End of FITAUPS.INC -)
```



## LISTING FOR FILE C:\PASCAL\INC\VIEWSIGS.INC

```

1:  {- This file (VIEWSIGS.INC) defines the procedure for showing the realtime -}
2:  {- PSD outputs on the screen, together with the determined amplitude and -}
3:  {- phase data. This is used for optimising the level of amplification of -}
4:  {- the PM tube output. -}
5:
6:  Procedure ViewPSDSignals;
7:
8:  Type SignalBox = Record
9:      x1, y1, x2, y2 : word;
10:  End;
11:
12:  Var
13:      PSD0, PSD1, Amplitude, Phase : extended;
14:      loop : word;
15:      PSD0Box, PSD1Box, AmplitudeBox, PhaseBox : SignalBox;
16:
17:  Begin
18:      With PSD0Box do begin
19:          x1 := 1; y1 := 15; x2 := 250; y2 := 65;
20:      end;
21:      With PSD1Box do begin
22:          x1 := 300; y1 := 15; x2 := 550; y2 := 65;
23:      end;
24:      With AmplitudeBox do begin
25:          x1 := 1; y1 := 85; x2 := 250; y2 := 135;
26:      end;
27:      With PhaseBox do begin
28:          x1 := 300; y1 := 85; x2 := 550; y2 := 135;
29:      end;
30:      SetGraphMode(CG4Hi);
31:      SetTextJustify(CenterText, TopText);
32:      OutTextXY(275,0,'REAL TIME RAW DATA');
33:      With PSD0Box do Rectangle(x1-1, y1-1, x2+1, y2+1);
34:      With PSD1Box do Rectangle(x1-1, y1-1, x2+1, y2+1);
35:      With AmplitudeBox do Rectangle(x1-1, y1-1, x2+1, y2+1);
36:      With PhaseBox do Rectangle(x1-1, y1-1, x2+1, y2+1);
37:      OutTextXY(125, 71, 'PSD 0 Output');
38:      OutTextXY(425, 71, 'PSD 1 Output');
39:      OutTextXY(125, 141, 'Amplitude');
40:      OutTextXY(425, 141, 'Phase');
41:      OutTextXY(275, 180, 'Press any key to exit');
42:      Repeat
43:          Begin
44:              for loop := 0 to 249 do begin
45:                  AD_Value(0, PSD0);
46:                  AD_Value(1, PSD1);
47:                  Amplitude := Sqrt(PSD0 * PSD0 + PSD1 * PSD1);
48:                  If PSD1 <> 0 then Phase := ArcTan(Abs(PSD0 / PSD1))
49:                      else Phase := pi/2;
50:                  If PSD0 * PSD1 < 0 then Phase := Pi - Phase;
51:                  If PSD0 < 0 then Phase := Phase + Pi;
52:                  Phase := 2 * pi - Phase;
53:                  With PSD0Box do PutPixel(x1 + loop, y2 - Trunc((PSD0 + 10) * 2.5), 7);
54:                  With PSD1Box do PutPixel(x1 + loop, y2 - Trunc((PSD1 + 10) * 2.5), 7);
55:                  With AmplitudeBox do PutPixel(x1 + loop, y2 - Trunc(Amplitude*50/14), 7);

```



```

56:      With PhaseBox do PutPixel(x1 + loop, y2 - Trunc(Phase * 7.9577),?);
57:  end;
58:  With PSD08Box do Begin
59:      SetViewPort(x1, y1, x2, y2, True);
60:      ClearViewPort;
61:  End;
62:  With PSD18Box do Begin
63:      SetViewPort(x1, y1, x2, y2, True);
64:      ClearViewPort;
65:  End;
66:  With AmplitudeBox do Begin
67:      SetViewPort(x1, y1, x2, y2, True);
68:      ClearViewPort;
69:  End;
70:  With PhaseBox do Begin
71:      SetViewPort(x1, y1, x2, y2, True);
72:      ClearViewPort;
73:  End;
74:  SetViewPort(0,0,639,199,True);
75:  End;
76:  Until KeyPressed;
77:  If ReadKey = ' ' then begin end;
78:  TextMode(3);
79:  End; {- of ViewPSDSignals -}
80:
81:  {- End of VIEWSI6S.INC -}

```

## LISTING FOR FILE C:\PASCAL\INC\RUNEXP.INC

```

1:  {- This file (RUNEXP.INC) contains the procedures used whilst the AUPS    -}
2:  {- software is fetching data from the phase-sensitive detectors.          -}
3:
4:  Procedure RunExperiment(QPrompts : boolean; CurrentRun : word);
5:
6:  Label AbortRun;
7:
8:  Var
9:      Rejected, Main_loop, loop, loop2, error : word;
10:     MeanAmp, RMSAmp : longint;
11:     Av_Amp : extended;
12:     SubTotal, PhaseOfs : longint;
13:     Phase, Amplitude: QuickArray;
14:     CorrelPlot, CorrelChan, AdjPhase, DPhase, Histogram,
15:     AWPData, AWPData : LongIntArray;
16:     Dummy : Char;
17:     Raw_Data : text;
18:     DispHistogram, Toggle, Weighting : boolean;
19:     tau, t : word;
20:     integral, temp1, temp2 : longint;
21:
22:  Begin
23:     ClearUserPrompt;
24:     UserPrompt('Fetching signal processing parameters from 5210...');
25:     Fetch5210Data;
26:     DataFileName := 'NOTNAMED.YET';
27:     If QPrompts then CurrentVolume := '';
28:     Data.Date := DateStr;
29:     Data.Time := TimeStr;
30:     Ext_Clock := False;
31:     Weighting := True;
32:     Toggle := False;
33:     DispHistogram := True;
34:     If Data.ElectrodeAmp = 0 then Ext_Irig := False else Ext_Irig := True;
35:     Rejected := 0;
36:     for loop := 0 to Data.NumPoints - 1 do AWPData[loop] := 0;
37:     for loop := 0 to Data.NumPoints - 1 do AWPData[loop] := 0;
38:     for loop := 0 to Data.NumPoints - 1 do CorrelPlot[loop] := 0;
39:     Data.AmpCorrelationFactor := 0;
40:     RMSAmp := 0;
41:     MeanAmp := 0;
42:     for loop := 0 to 150 do Histogram[loop] := 0;
43:     {- Fetch a dummy batch of data to set up DMA arrays -}
44:     Repeat
45:         FetchData(Phase, Amplitude, Histogram, CorrelChan,
46:             Av_Amp, Data.NumPoints + 2, Data.DelayTime, True);
47:         Wait_for_DMA;
48:         If Status = 1 then begin
49:             ErrorBeep;
50:             GotoXY(1,25);
51:             TextColor(Yellow);
52:             TextBackGround(Red);
53:             Write('CRITICAL ERROR! DMA Time-out. Check electrode drive is ON. SPACEBAR to cont. ');
54:             Repeat Until ReadKey = ' ';
55:             TextColor(White);

```

```

56:         TextBackground(Black);
57:         ClearUserPrompt;
58:     end;
59:     Until Status <> 1;
60:     UserPrompt('Acquiring and processing data...please wait. ');
61:     for Main_loop := 1 to Data.NumSamples do
62:         Begin
63:             FetchData(Phase, Amplitude, Histogram, CorrelChan,
64:                 Av_Amp, Data.NumPoints + 2, Data.DelayTime, false);
65:             GotoXY(46,25);
66:             Write('Run: ',CurrentRun,' Sample: ',Main_loop,' ');
67:             {- Remove discontinuities in the phase data occurring at 0/360 degree -}
68:             {- crossovers, which would otherwise give false phase change data. -}
69:
70:             PhaseOfs := 0;
71:             for loop := 0 to Data.Numpoints do begin
72:                 AdjPhase[loop] := Phase[loop] + PhaseOfs;
73:                 If Phase[loop+1]-Phase[loop] <= - 160 then PhaseOfs := PhaseOfs+256
74:                 else
75:                     If Phase[loop+1]-Phase[loop] >= 160 then PhaseOfs := PhaseOfs-256;
76:             end;
77:             AdjPhase[Data.Numpoints + 1] := Phase[Data.Numpoints + 1] + PhaseOfs;
78:             for loop := 1 to Data.NumPoints + 1 do
79:                 DPhase[loop] := AdjPhase[loop]-AdjPhase[loop-1];
80:
81:             { ----- }
82:             { This section calculates }
83:             { }
84:             {  $AWPD[\tau] := Q(\tau) = \int_0^{\tau} A(t) \langle t \rangle dt$  }
85:             { }
86:             { }
87:             { and }
88:             { }
89:             { }
90:             {  $AWPS[\tau] := Q(\tau) = \int_0^{\tau} A(t) \langle t \rangle^2 dt$  }
91:             { }
92:             { }
93:             { }
94:             { using the Trapezoidal method for approximating definite integrals. }
95:             { ----- }
96:
97:
98:             for tau := 0 to Data.NumPoints - 1 do begin
99:                 temp1 := Amplitude[tau];
100:                 MeanAmp := MeanAmp + temp1;
101:                 RMSAmp := RMSAmp + temp1 * temp1;
102:             end;
103:             integral := 0;
104:             for tau := 1 to Data.NumPoints - 1 do begin
105:                 integral := integral + Amplitude[tau] * DPhase[tau];
106:                 AWPDData[tau] := AWPDData[tau] + integral;
107:                 AWPSData[tau] := AWPSData[tau] + (integral div 256) * (integral div 256);
108:             end;
109:             {- Calculate and update correlation function -}
110:             for loop := 0 to Data.NumPoints - 1 do

```

```

111:      CorrelPlot[loop] := CorrelPlot[loop]
112:      * CorrelChan[0] * CorrelChan[loop];
113:      If KeyPressed then if ReadKey = 'X' then goto AbortRun;
114:      {- Wait for alternate batch of data to be transferred -}
115:      Wait_for_DMA;
116:      If Status = 1 then begin
117:          ErrorBeep;
118:          ClearUserPrompt;
119:          GotoXY(1,25);
120:          TextColor(Yellow);
121:          TextBackGround(Red);
122:          Write('FATAL ERROR! DMA Time-out. Run aborted. Press SPACEBAR. ');
123:          Repeat Until ReadKey = ' ';
124:          TextColor(White);
125:          TextBackGround(Black);
126:          Goto AbortRun;
127:      end;
128:      end; {- of Main_loop -}
129:      AbortRun:
130:      ClearUserPrompt;
131:      UserPrompt('Normalising data...');
132:      for loop := 0 to Data.NumPoints - 1 do begin
133:          Data.AWPDData[loop] := AWPDData[loop];
134:          Data.AWPDData[loop] := Data.AWPDData[loop] /
135:              (MeanAmp / Data.NumPoints) * pi / 128.0;
136:          Data.AWPSData[loop] := AWPSData[loop];
137:          Data.AWPSData[loop] := Data.AWPSData[loop] / (RMSAmp / Data.NumPoints)
138:              * Sqr(2 * pi); {- Since already div by 256*256 -}
139:          Data.TimeData[loop] := loop;
140:          Data.TimeData[loop] := Data.TimeData[loop] * Data.DelayTime;
141:      end;
142:      for loop := 0 to 150 do Data.Histogram[loop] := Histogram[loop];
143:      for loop := Data.NumPoints - 1 downto 0 do
144:          Data.AmpCorr[loop] := CorrelPlot[loop] / CorrelPlot[0];
145:      End; {- of RunExperiment -}
146:
147:      {- End of RUNEXP.INC -}

```

## LISTING FOR FILE C:\PASCAL\INC\DOSPROCS.INC

```

1:  {- This file (DOSPROCS.INC) contains the various routines which are DOS-  -}
2:  {- related such as file access/data storage.                             -}
3:
4:  Function Exist(Name : String) : boolean;
5:
6:  Var
7:    DirInfo : SearchRec;
8:
9:  Begin
10:    FindFirst(Name, Archive, DirInfo);
11:    If (DosError = 0) and (DirInfo.Attr < 64) then Exist := True else Exist := False;
12:  End; {- of Exist -}
13:
14:  Function OverWrite(Name : String) : boolean;
15:
16:  Begin
17:    OverWrite := False;
18:    GotoXY(1,25);
19:    ErrorBeep;
20:    TextColor(Yellow);
21:    TextBackGround(Red);
22:    Write('WARNING! ',Name,' already exists - overwrite? Y or N');
23:    If UpCase(ReadKey) = 'Y' then OverWrite := True;
24:    TextColor(White);
25:    TextBackGround(Black);
26:  End; {- of OverWrite -}
27:
28:  Procedure WriteParamsToDisc;
29:
30:  Var DataFile : File of DataRecord;
31:      Name : String;
32:
33:  Begin
34:    UserPrompt('Enter DOS path to write parameters to: ');
35:    DefFuncKey(DataFileName);
36:    ReadLn(Name);
37:    DisplayParameterMenu;
38:    ClearUserPrompt;
39:    UserPrompt('Wait...');
40:    If Exist(Name) then If Not(OverWrite(Name)) then exit;
41:    Assign(DataFile, Name);
42:    Rewrite(DataFile);
43:    If CheckforIOError then exit;
44:    ClearUserPrompt;
45:    UserPrompt('Writing parameters to '+Name);
46:    Write(DataFile, Data);
47:    Close(DataFile);
48:  End; {- of WriteParamsToDisc -}
49:
50:  Procedure ReadParamsFromDisc;
51:
52:  Var DataFile : File of DataRecord;
53:      Name : String;
54:
55:  Begin

```



```

56:   ClearUserPrompt;
57:   UserPrompt('Enter DOS path to read parameters from: ');
58:   DefFuncKey(DataFileName);
59:   Readln(Name);
60:   DisplayParameterMenu;
61:   ClearUserPrompt;
62:   UserPrompt('Wait...');
63:   Assign(DataFile, Name);
64:   Reset(DataFile);
65:   If CheckforIOError then Exit;
66:   ClearUserPrompt;
67:   UserPrompt('Reading parameters from '+Name);
68:   Read(DataFile, Data);
69:   If CheckforIOError then Exit;
70:   Close(DataFile);
71: End; (- of ReadParamsFromDisc -)
72:
73: Procedure DirList(Attribute : Byte);
74:
75:   Var
76:     SearchRecord : SearchRec;
77:     CurrentDir   : String;
78:
79:
80:   Begin
81:     ClrScr;
82:     DisplayMainTitle;
83:     GotoXY(1,3);
84:     FindFirst('\*. *', VolumeID, SearchRecord);
85:     CurrentVolume := SearchRecord.Name;
86:     Delete(CurrentVolume, Pos('.', CurrentVolume), 1);
87:     TextColor(DarkGray);
88:     Write('VOLUME : '); TextColor(Red) ; Writeln(CurrentVolume);
89:     TextColor(DarkGray);
90:     Write('CAPACITY: '); TextColor(Red) ; Writeln(DiskSize(0), ' bytes');
91:     TextColor(DarkGray);
92:     Write('FREE   : '); TextColor(Red) ; Writeln(DiskFree(0), ' bytes');
93:     TextColor(DarkGray);
94:     GetDir(0, CurrentDir);
95:     Write('PATH   : '); TextColor(Red) ; Writeln(CurrentDir);
96:     Writeln('');
97:     TextColor(Green);
98:     FindFirst('*. *', Attribute, SearchRecord);
99:     While DosError = 0 do
100:       begin
101:         If WhereX > 66 then Writeln('');
102:         If (Attribute <> 16) or ((Attribute = 16) and (SearchRecord.Attr = 16))
103:           then begin
104:             Write(SearchRecord.Name, ' ');
105:             GotoXY(16 * (1 + WhereX div 16), WhereY);
106:           end;
107:           FindNext(SearchRecord);
108:         end;
109:       End; (- of DirList -)
110:

```

```

111: Procedure WriteDataToDisc(fileName : String);
112:
113: Begin
114:   If fileName = '' then
115:     begin
116:       DirList(Archive);
117:       UserPrompt('Enter file to write data to: ');
118:       DefFuncKey(DataFileName);
119:       Readln(DataFileName);
120:       If DataFileName = '' then Exit;
121:       DisplayDataStorageMenu;
122:     end
123:     else DataFileName := fileName;
124:   ClearUserPrompt;
125:   UserPrompt('Wait...');
126:   If Exist(DataFileName) then If Not(OverWrite(DataFileName)) then exit;
127:   Assign(DataFile, DataFileName);
128:   If CheckforIOError then Exit;
129:   Rewrite(DataFile);
130:   If CheckforIOError then Exit;
131:   ClearUserPrompt;
132:   UserPrompt('Writing data to '+DataFileName);
133:   Write(DataFile, Data);
134:   Close(DataFile);
135:   If CheckforIOError then Exit;
136: End; {- of WriteDataToDisc -}
137:
138: Procedure WriteGrapherFileToDisc;
139:
140: Var loop : word;
141:   OutputFile : Text;
142:
143: Begin
144:   DirList(Archive);
145:   UserPrompt('Enter file to write data to: ');
146:   DefFuncKey(DataFileName);
147:   Readln(DataFileName);
148:   If DataFileName = '' then Exit;
149:   DisplayDataStorageMenu;
150:   ClearUserPrompt;
151:   UserPrompt('Wait...');
152:   If Exist(DataFileName) then If Not(OverWrite(DataFileName)) then exit;
153:   Assign(OutputFile, DataFileName);
154:   If CheckforIOError then Exit;
155:   Rewrite(OutputFile);
156:   If CheckforIOError then Exit;
157:   ClearUserPrompt;
158:   UserPrompt('Writing data to '+DataFileName);
159:   Writeln(OutputFile, 'TR 0.000 2.000');
160:   Writeln(OutputFile, 'SC 0.600 0.600');
161:   Writeln(OutputFile, 'SS "DEFAULT.SYM"');
162:   Writeln(OutputFile, 'SP 2');
163:   Write(OutputFile, 'PS 1.000 -2.300 0.200 0.000 ');
164:   Writeln(OutputFile, '"Electrophoretic mobility: ', GlobalMobility:11, ' m2/Us"');
165:   Write(OutputFile, 'PS 1.000 -2.000 0.200 0.000 ');

```

```

166:   Writeln(Outputfile,"Electrode: ',Data.ElectrodeAmp:1:2,' U at ',Data.ElectrodeFreq:1:2,' Hz across '
167:         ,Data.ElectrodeGap:1:2,' mm");
168:   Write(Outputfile,'PS 1.000 -1.700 0.200 0.000 ');
169:   Writeln(Outputfile,"Sampling: ',Data.NumSamples,' samples of ',Data.NumPoints,' ',Data.DelayTime:1:6,' s chan
170:   Write(Outputfile,'PS 1.000 -1.400 0.200 0.000 ');
171:   Write(Outputfile,"Scattering angle: ',Data.Angle:1:2,' deg  ');
172:   Writeln(Outputfile,"Temperature: ',Data.Temperature:1:1,' deg C");
173:   Write(Outputfile,'PS 1.000 -1.100 0.200 0.000 ');
174:   Write(Outputfile,"Sample: ',Data.SampleName);
175:   Writeln(Outputfile,' Date: ',Data.Date,' Time: ',Data.Time,' ');
176:   Writeln(Outputfile,'SC 1.667 1.667');
177:   Writeln(Outputfile,'IR 0.000 -2.000');
178:   for loop := 0 to 149 do
179:     If loop <= Data.NumPoints - 1 then
180:       Writeln(Outputfile, (Data.DelayTime * loop):11,' ',Data.Histogram[loop]:11,
181:         ' ',Data.AWPSData[loop]:11,' ',Data.AWPDData[loop]:11,' ',
182:         Data.AmpCorr[loop]:11)
183:     else
184:       Writeln(Outputfile, (loop * Data.DelayTime):11,' ',Data.Histogram[loop]:11);
185:   Close(Outputfile);
186:   If CheckforIOError then Exit;
187: End; {- of WriteGrapherfiletoDisc -}
188:
189: Procedure ReadDatafromDisc;
190:
191: Begin
192:   DirList(Archive);
193:   ClearUserPrompt;
194:   UserPrompt('Enter file to read data from: ');
195:   DefFuncKey(DataFileName);
196:   Readln(DataFileName);
197:   If DataFileName = '' then Exit;
198:   DisplayDataStorageMenu;
199:   ClearUserPrompt;
200:   UserPrompt('Wait...');
201:   Assign(DataFile, DataFileName);
202:   If CheckforIOError then Exit;
203:   Reset(DataFile);
204:   If CheckforIOError then Exit;
205:   ClearUserPrompt;
206:   UserPrompt('Reading data from '+DataFileName);
207:   Read(DataFile, Data);
208:   Close(DataFile);
209:   If CheckforIOError then Exit;
210: End; {- of ReadDatafromDisc -}
211:
212: Procedure ChangeDirectory;
213:
214: Begin
215:   DirList(Directory);
216:   ClearUserPrompt;
217:   UserPrompt('Enter new directory: ');
218:   DefFuncKey(CurrentDirectory);
219:   Readln(CurrentDirectory);
220:   If CurrentDirectory = '' then Exit;

```

221:     DisplayDataStorageMenu;  
222:     ChDir(CurrentDirectory);  
223:     If CheckforIOError then Exit;  
224:     End; {- of ChangeDirectory -}  
225:  
226:     {- End of DOSPROCS.INC -}

## LISTING FOR FILE C:\PASCAL\INC\MULTIRUN.INC

```

1:  (- This file (MULTIRUN.INC) contains the procedures for implementing the  -)
2:  (- 'Multi-Run' facility.                                                -)
3:
4:  Procedure CalibrateAutoTrack;
5:
6:  Var
7:    LowVolts, HighVolts, LowFreq, HighFreq : extended;
8:
9:  Begin
10:    UserPrompt('Calibrating AutoTrack. Please wait. ');
11:    CentreOffset := 0;
12:    LowVolts := -0.1;
13:    HighVolts := +0.1;
14:    DA_Value(0, LowVolts);
15:    Fetch5210Data;
16:    LowFreq := Data.LIARefFreq / 1000;
17:    DA_Value(0, HighVolts);
18:    Fetch5210Data;
19:    HighFreq := Data.LIARefFreq / 1000;
20:    FreqToVoltsFactor := (HighFreq - LowFreq) / (HighVolts - LowVolts);
21:    DA_Value(0, CentreOffset);
22:  End; (- of CalibrateAutoTrack -)
23:
24:  Procedure EnterMultiRunMenu;
25:
26:  Var
27:    NumRunsStr: String;
28:
29:  Procedure DisplayMultiRunMenu;
30:
31:  Begin
32:    ClrScr;
33:    DisplayMainTitle;
34:    Str(NumRuns, NumRunsStr);
35:    MenuTitle('MULTI-RUN MENU');
36:    MenuItem(1, 'Long form output file ', LongName);
37:    MenuItem(2, 'Short form output file ', ShortName);
38:    MenuItem(3, 'Number of runs ', NumRunsStr);
39:    MenuItem(4, 'Begin', '');
40:    If AutoTrack then MenuItem(5, 'Disable AutoTrack (Currently ON)', '');
41:    else MenuItem(5, 'Enable AutoTrack (Currently OFF)', '');
42:    MenuItem(6, 'Calibrate AutoTrack', '');
43:    MenuItem(7, 'Main Menu', '');
44:    UserPrompt('Make Selection (1-7)');
45:  End; (- of DisplayMultiRunMenu -)
46:
47:  Procedure InputLongName;
48:
49:  Begin
50:    Repeat
51:      Begin
52:        UserPrompt('Enter family file name for long form storage: ');
53:        DefFuncKey(LongName);
54:        Readln(LongName);
55:      End;

```



```

56:     Until IOResult = 0;
57: End; {- of InputLongName -}
58:
59: Procedure InputShortName;
60:
61: Begin
62:     Repeat
63:         Begin
64:             UserPrompt('Enter file name for short form storage: ');
65:             DefFuncKey(ShortName);
66:             Readln(ShortName);
67:             DefFuncKey(ShortName);    {- SO THAT F1 IS DEFINED FOR SHELL -}
68:         End;
69:     Until IOResult = 0;
70: End; {- of InputShortName -}
71:
72: Procedure InputNumberOfRuns;
73:
74: Begin
75:     Repeat
76:         Begin
77:             UserPrompt('Enter number of runs: ');
78:             DefFuncKey(NumRunsStr);
79:             Readln(NumRuns);
80:         End;
81:     Until (IOResult = 0) and (NumRuns > 0);
82: End; {- of InputNumberOfRuns -}
83:
84: Procedure DoMultiRun;
85:
86: Var
87:     Shortfile : Text;
88:     Multiloop : word;
89:     Suffix : String;
90:     Mob, Mob2, Vel, Vel2, Diam, Diam2, SN, SN2, Int, Int2, Dq, Dq2 : extended;
91:     SMob, SVel, SDiam, SSN, SInt, SDq : extended;
92:
93: Begin
94:     If Exist(ShortName) then If Not(OverWrite(ShortName)) then exit;
95:     ClearUserPrompt;
96:     UserPrompt('Wait...creating '+ShortName);
97:     Assign(ShortFile, ShortName);
98:     Rewrite(ShortFile);
99:     If CheckForIOError then Exit;
100:    Writeln(ShortFile,Data.Date,' ',Data.SampleName);
101:    Writeln(ShortFile,'APPLIED FIELD: ',ElectrodeAmpStr(Data.ElectrodeAmp),
102:            ' / ',ElectrodeGapStr(Data.ElectrodeGap),' at ',
103:            ElectrodefreqStr(Data.Electrodefreq));
104:    Writeln(ShortFile,'SAMPLING      : ',NumSamplesStr(Data.NumSamples),
105:            ' samples of ',NumPointsStr(Data.NumPoints),' ',
106:            DelayTimeStr(Data.DelayTime),' channels. ');
107:    Writeln(ShortFile,'');
108:    Writeln(ShortFile,' MOBILITY      VELOCITY    DIAMETER      S/N RATIO      AMPLITUDE CORRELN  TIME ');
109:    Writeln(ShortFile,' (m2/Vs)      (um/s)      (nm)              Q-factor DIAM/nm   ');
110:    Writeln(ShortFile,'');

```

```

111:   Close(ShortFile);
112:   Mob := 0 ; Mob2 := 0;
113:   Vel := 0 ; Vel2 := 0;
114:   Diam := 0 ; Diam2 := 0;
115:   SN := 0 ; SN2 := 0;
116:   Int := 0 ; Int2 := 0;
117:   Dq := 0 ; Dq2 := 0;
118:   for Multiloop := 1 to NumRuns do
119:   begin
120:     RunExperiment(NoPrompts, Multiloop);
121:     Str(Multiloop, Suffix);
122:     WriteDataToDisc(LongName + '.' + Suffix);
123:     EnterFitRUPSMenus(NoPrompts);
124:     ClearUserPrompt;
125:     UserPrompt('Writing data to short-form output file '+ShortName);
126:     Assign(ShortFile, ShortName);
127:     Append(ShortFile);
128:     If CheckForIOError then Exit;
129:     Writeln(ShortFile, ' ', GlobalMobility:11, ' ', GlobalVelocity:6:2,
130:              ' ', GlobalDiameter:6:1, ' ', GlobalSN:6:3,
131:              ' ', Abs(GlobalIntercept):8:5, ' ', GlobalGradient:6:1, ' ', Data.Time);
132:     Close(ShortFile);
133:     If KeyPressed then If ReadKey = 'X' then Exit;
134:     Mob := Mob + GlobalMobility; Mob2 := Mob2 + Sqr(GlobalMobility);
135:     Vel := Vel + GlobalVelocity; Vel2 := Vel2 + Sqr(GlobalVelocity);
136:     Diam := Diam + GlobalDiameter; Diam2 := Diam2 + Sqr(GlobalDiameter);
137:     SN := SN + GlobalSN ; SN2 := SN2 + Sqr(GlobalSN);
138:     Int := Int + Abs(GlobalIntercept) ; Int2 := Int2 + Sqr(GlobalIntercept);
139:     Dq := Dq + GlobalGradient ; Dq2 := Dq2 + Sqr(GlobalGradient);
140:   end;
141:   SMob := Sqrt(ABS(NumRuns * Mob2 - Sqr(Mob)) / Sqr(NumRuns));
142:   Mob := Mob / NumRuns;
143:   SVel := Sqrt(ABS(NumRuns * Vel2 - Sqr(Vel)) / Sqr(NumRuns));
144:   Vel := Vel / NumRuns;
145:   SDiam := Sqrt(ABS(NumRuns * Diam2 - Sqr(Diam)) / Sqr(NumRuns));
146:   Diam := Diam / NumRuns;
147:   SSN := Sqrt(ABS(NumRuns * SN2 - Sqr(SN)) / Sqr(NumRuns));
148:   SN := SN / NumRuns;
149:   SInt := Sqrt(ABS((NumRuns * Int2 - Sqr(Int))) / Sqr(NumRuns));
150:   Int := Int / NumRuns;
151:   SDq := Sqrt(ABS(NumRuns * Dq2 - Sqr(Dq)) / Sqr(NumRuns));
152:   Dq := Dq / NumRuns;
153:   Assign(ShortFile, ShortName);
154:   Append(ShortFile);
155:   Writeln(ShortFile, '-----');
156:   Writeln(ShortFile, ' ', Mob:11, ' ', Vel:6:2, ' ', Diam:6:1, ' ',
157:            ' ', SN:6:3, ' ', Int:8:5, ' ', Dq:5:1);
158:   Writeln(ShortFile, ' ', SMob:11, ' ', SVel:6:2, ' ', SDiam:6:1, ' ',
159:            ' ', SSN:6:3, ' ', SInt:8:5, ' ', SDq:6:1);
160:   Writeln(ShortFile, '');
161:   Close(ShortFile);
162:   ErrorBeep;
163: End; (- of DoMultiRun -)
164:
165: Begin

```

```

166:   Repeat
167:     Begin
168:       DisplayMultiRunMenu;
169:       Case ReadKey of
170:         '1' : InputLongName;
171:         '2' : InputShortName;
172:         '3' : InputNumberOfRuns;
173:         '4' : DoMultiRun;
174:         '5' : Begin
175:           AutoTrack := Not(AutoTrack);
176:           If Not(AutoTrack) then
177:             begin
178:               CentreOffSet := 0;
179:               DA_Value(0, CentreOffSet);
180:             end;
181:           end;
182:         '6' : CalibrateAutoTrack;
183:         '7' : Exit;
184:       else
185:         ErrorBeep;
186:       end;
187:     end;
188:   Until False;
189: End; {- of EnterMultiRunMenu -}
190:
191: {- End of MULTIRUN.INC -}

```

## LISTING FOR FILE C:\PASCAL\INC\ENTMENU2.INC

```

1:      {- This file (ENTMENU2.INC) contains some further menu control routines.  -}
2:
3:      Procedure EnterParameterMenu;
4:
5:      Begin
6:          Repeat Begin
7:              DisplayParameterMenu;
8:              Case ReadKey of
9:                  '1' : EnterOpticalMenu;
10:                 '2' : EnterSignalProcessingMenu;
11:                 '3' : EnterElectrodeDriveMenu;
12:                 '4' : EnterOtherMenu;
13:                 '5' : WriteParamsToDisc;
14:                 '6' : ReadParamsFromDisc;
15:                 '7' : Exit;
16:             else
17:                 ErrorBeep;
18:             end;
19:         end;
20:     Until False;
21: End; {- of EnterParameterMenu -}
22:
23: Procedure EnterDataStorageMenu;
24:
25: Begin
26:     Repeat Begin
27:         DisplayDataStorageMenu;
28:         Case ReadKey of
29:             '1' : ReadDataFromDisc;
30:             '2' : WriteDataToDisc('');
31:             '3' : WriteGrapherFileToDisc;
32:             '4' : ChangeDirectory;
33:             '5' : Exit;
34:         else
35:             ErrorBeep;
36:         end;
37:     end;
38: Until False;
39: End; {- of EnterDataStorageMenu -}
40:
41: Procedure EnterDataProcessingMenu;
42:
43: Const
44:     SemiLog : Boolean = True;
45:     Normal  : Boolean = False;
46:
47: Begin
48:     Repeat Begin
49:         DisplayDataProcessingMenu;
50:         Case ReadKey of
51:             '1' : EnterFitAJPMenu(Prompts);
52:             '2' : DisplayHistogram(Data.Histogram);
53:             '3' : DisplayCorrelogram(Data.AmpCorr, Normal);
54:             '4' : DisplayCorrelogram(Data.AmpCorr, SemiLog);
55:             '5' : Exit;

```

```
56:     else
57:         ErrorBeep;
58:     end;
59:     end;
60:     Until False;
61: End; {- of EnterDataProcessingMenu -}
62:
63: {- End of ENTMENU2.INC -}
```



## LISTING FOR FILE C:\PASCAL\UNIT\DT2818.PAS

```

1:  Unit DT2818;
2:
3:  {$N+}
4:
5:  Interface
6:
7:  Uses Dos;
8:
9:  Const
10:
11:      Base_Address : Word = $2EC;
12:      Command_Reg  : Word = $2ED;
13:      Status_Reg   : Word = $2ED;
14:      Data_Reg     : Word = $2EC;
15:      Command_Wait : Byte = $4;
16:      Write_Wait   : Byte = $2;
17:      Read_Wait    : Byte = $5;
18:      CReset       : Byte = $0;
19:      CClear        : Byte = $1;
20:      CError        : Byte = $2;
21:      CClock        : Byte = $3;
22:      CSin          : Byte = $4;
23:      CSout         : Byte = $5;
24:      CIOIn         : Byte = $6;
25:      CIOOut        : Byte = $7;
26:      CDAout        : Byte = $8;
27:      CSOA          : Byte = $9;
28:      CUDOA         : Byte = $A;
29:      CTest         : Byte = $B;
30:      CADin         : Byte = $C;
31:      CSAO          : Byte = $D;
32:      CRAO          : Byte = $E;
33:      CStop         : Byte = $F;
34:      CDMA          : Byte = $10;
35:      CContinuous   : Byte = $20;
36:      CExt_Clock    : Byte = $40;
37:      CExt_Irig     : Byte = $80;
38:      Base_Factor   : Extended = 4096.0;
39:      Base_Channels : Byte = 4;
40:      Gain_Code     : Byte = 0;
41:      Top_Range     : Extended = 10.0;
42:      Bottom_Range  : Extended = -10.0;
43:      Top_Freq      : Extended = 800000.0;
44:      Base_Rate     : Extended = 1.25e-06;
45:      DMA_Addr_1    : Byte = $2;
46:      DMA_Byte_1    : Byte = $3;
47:      DMA_Addr_3    : Byte = $6;
48:      DMA_Byte_3    : Byte = $7;
49:      DMA_Mask      : Byte = $A;
50:      DMA_Mode      : Byte = $B;
51:      DMA_Clr_BP    : Byte = $C;
52:      DMA_Page_1    : Byte = $83;
53:      DMA_Page_3    : Byte = $82;
54:      Write_1       : Byte = $45;
55:      Write_3       : Byte = $47;

```



```

111:   Function Voltage(Analog : word) : extended;
112:   Function Seconds(Hour, Min, Sec, Sec100 : word) : extended;
113:   Function DMA_Count(Channel : byte) : word;
114:   Function Usable_Length(var ArrayPtr : DArrayPtr) : word;
115:
116:
117:   Implementation
118:
119:   Procedure Disable_Sys_Clock;
120:
121:   Begin
122:     Disable_Interrupts;
123:     Port[33] := Port[33] or 1;
124:     Enable_Interrupts;
125:   End; {- of Disable_Sys_Clock -}
126:
127:   Procedure Enable_Sys_Clock;
128:
129:   Begin
130:     Disable_Interrupts;
131:     Port[33] := Port[33] and 254;
132:     Enable_Interrupts;
133:   End; {- of Enable_Sys_Clock -}
134:
135:
136:   Procedure Get_Page_and_Offset;
137:
138:   Var
139:     Segment, Offset : word;
140:     Abs_Address, Temp_Page, Temp_Offs : longint;
141:
142:   Begin
143:
144:     Segment := Seg(ArrayPtr^);
145:     Offset  := Of(ArrayPtr^);
146:     Abs_Address := Segment;
147:     Abs_Address := Abs_Address * 16 + Offset;
148:     Temp_Page := Abs_Address div 65536;
149:     Temp_Offs := Abs_Address mod 65536;
150:     Page := Temp_Page;
151:     Pg_Offset := Temp_Offs;
152:
153:   End; {- of Get_Page_and_Offset -}
154:
155:
156:
157:
158:   Function Seconds;
159:
160:   Var T1,T2,T3,T4 : extended;
161:
162:   Begin
163:     T1 := Hour;
164:     T2 := Min;
165:     T3 := Sec;

```

```

166:     T4 := Sec100;
167:     Seconds := (T1 * 3600) + (T2 * 60) + T3 + (T4 / 100);
168:     End; {- of Seconds -}
169:
170:     Function Usable_Length;
171:
172:     Var
173:         Offset : Word;
174:         Page : Byte;
175:
176:     Begin
177:
178:         Get_Page_and_Offset(ArrayPtr, Page, Offset);
179:         If Offset = 0 then Usable_Length := 0
180:             else Usable_Length := 65535 - Offset + 1;
181:
182:     End; {- of Usable_Length -}
183:
184:
185:     Procedure ClearDT;
186:
187:     Var Temp : byte;
188:
189:     Begin
190:
191:         Port[Command_Reg] := CStop;
192:         Temp := Port[Data_Reg];
193:         While Port[Status_Reg] and Write_Wait <> 0 do;
194:             While Port[Status_Reg] and Command_Wait = 0 do;
195:                 Port[Command_Reg] := CClear;
196:             End; {- of ClearDT -}
197:
198:
199:
200:     Procedure ResetDT;
201:
202:     Var Temp : byte;
203:
204:     Begin
205:
206:         Port[Command_Reg] := CStop;
207:         Temp := Port[Data_Reg];
208:         While Port[Status_Reg] and Write_Wait <> 0 do;
209:             While Port[Status_Reg] and Command_Wait = 0 do;
210:                 Port[Command_Reg] := CReset;
211:                 Temp := Port[Data_Reg];
212:             End; {- of ResetDT -}
213:
214:
215:     Procedure StopDT;
216:
217:     Var Temp : byte;
218:
219:     Begin
220:

```

```

221:     Port[Command_Reg] := CStop;
222:     Temp := Port[Data_Reg];
223:
224: End; (- of StopDT -)
225:
226:
227: Procedure TestDT;
228:
229: Var
230:
231:     Temp : byte;
232:     Loop : integer;
233:
234: Begin
235:
236:     While Port[Status_Reg] and Write_Wait <> 0 do;
237:     While Port[Status_Reg] and Command_Wait = 0 do;
238:     Port[Command_Reg] := CTest;
239:     Error_Count := 0;
240:     for loop := 1 to 255 do
241:     begin
242:         while Port[Status_Reg] and Read_Wait = 0 do;
243:         temp := Port[Data_Reg];
244:         if temp <> loop then Error_Count := Error_Count + 1;
245:     end;
246:     Port[Command_Reg] := CStop;
247:     Temp := Port[Data_Reg];
248:
249: End; (- of TestDT -)
250:
251:
252:
253: Procedure Get_DT_Error;
254:
255: Var
256:     Err1, Err2 : byte;
257:
258: Begin
259:
260:     Status := Port[Status_Reg];
261:     If Status and $80 = 0 then begin Error:=0 ; Exit; End;
262:     StopDT;
263:     While Port[Status_Reg] and Write_Wait <> 0 do;
264:     While Port[Status_Reg] and Command_Wait = 0 do;
265:     Port[Command_Reg] := CError;
266:     While Port[Status_Reg] and Read_Wait = 0 do;
267:     Err1 := Port[Data_Reg];
268:     While Port[Status_Reg] and Read_Wait = 0 do;
269:     Err2 := Port[Data_Reg];
270:     Error := Err1 + 256 * Err2;
271:
272: End; (- of Get_DT_Error -)
273:
274:
275: Procedure Print_Error_Msg;

```



```

276:
277:   Begin
278:
279:     Write('*** DT2818 error ');
280:     If Error and 1 <> 0 then Writeln('0, Reserved');
281:     If Error and 2 <> 0 then Writeln('1, Command Overwrite Error');
282:     If Error and 4 <> 0 then Writeln('2, Clock Set Error');
283:     If Error and 8 <> 0 then Writeln('3, Digital Port Select Error');
284:     If Error and 16 <> 0 then Writeln('4, Digital Port Set Error');
285:     If Error and 32 <> 0 then Writeln('5, DAC Select Error');
286:     If Error and 64 <> 0 then Writeln('6, DAC Clock Error');
287:     If Error and 128 <> 0 then Writeln('7, DAC #Conversions Error');
288:     If Error and 256 <> 0 then Writeln('8, A/D Channel Error');
289:     If Error and 512 <> 0 then Writeln('9, A/D Gain Error');
290:     If Error and 1024 <> 0 then Writeln('10, A/D Clock Error');
291:     If Error and 2048 <> 0 then Writeln('11, A/D Multiplexer Error');
292:     If Error and 4096 <> 0 then Writeln('12, A/D #Conversions Error');
293:     If Error and 8192 <> 0 then Writeln('13, Data where Command Expected Error');
294:     If Error and 16384 <> 0 then Writeln('14, Reserved');
295:     If Error and 32768 <> 0 then Writeln('15, Reserved');
296:     If Error = 0 then Writeln('-1, No Error');
297:
298:   End; {- of Print_Error_Msg -}
299:
300:
301:   Procedure Set_DIO;
302:
303:   Begin
304:     While Port[Status_Reg] and Write_Wait <> 0 do;
305:     While Port[Status_Reg] and Command_Wait = 0 do;
306:       If Direction=Digital_In then
307:         If Not(Ext_Trig) then Port[Command_Reg]:=CSin
308:           else Port[Command_Reg]:=CSin + CExt_Trig;
309:       If Direction=Digital_Out then
310:         If Not(Ext_Trig) then Port[Command_Reg]:=CSout
311:           else Port[Command_Reg]:=CSout + CExt_Trig;
312:     While Port[Status_Reg] and Write_Wait <> 0 do;
313:     Port[Data_Reg] := Port_Select;
314:   End; {- of Set_DIO -}
315:
316:   Procedure Read_DIO;
317:
318:   Begin
319:     While Port[Status_Reg] and Write_Wait <> 0 do;
320:     While Port[Status_Reg] and Command_Wait = 0 do;
321:       If Ext_Trig then Port[Command_Reg] := CDIOIn + CExt_Trig
322:         else Port[Command_Reg] := CDIOIn;
323:     While Port[Status_Reg] and Write_Wait <> 0 do;
324:     Port[Data_Reg] := Port_Select;
325:     While Port[Status_Reg] and Read_Wait = 0 do;
326:     Result := Port[Data_Reg];
327:   End; {- of Read_DIO -}
328:
329:   Procedure Write_DIO;
330:

```

```

331: Begin
332:   While Port[Status_Reg] and Write_Wait <> 0 do;
333:     While Port[Status_Reg] and Command_Wait = 0 do;
334:       If Ext_Trig then Port[Command_Reg] := CIOOut + CExt_Trig
335:         else Port[Command_Reg] := CIOOut;
336:       While Port[Status_Reg] and Write_Wait <> 0 do;
337:         Port[Data_Reg] := Port_Select;
338:       While Port[Status_Reg] and Write_Wait <> 0 do;
339:         Port[Data_Reg] := Data;
340:     End; (- of Write_DIO -)
341:
342:
343:
344:
345:
346: Procedure Set_Clock_Period;
347:
348: Var
349:
350:   Ticks      : word;
351:   High, Low, Temp : byte;
352:
353: Begin
354:
355:   if (Period < (Base_Rate * 2.0)) or (Period > (Base_Rate * 65535.0))
356:     then Actual_Period := -10 (- ie invalid period -)
357:   else
358:     begin
359:       Ticks := Trunc( Period / Base_Rate );
360:       Actual_Period := Ticks * Base_Rate;
361:       While Port[Status_Reg] and Write_Wait <> 0 do;
362:         While Port[Status_Reg] and Command_Wait = 0 do;
363:           Port[Command_Reg] := CClock;
364:           High := Ticks div 256;
365:           Low  := Ticks mod 256;
366:           While Port[Status_Reg] and Write_Wait <> 0 do;
367:             Port[Data_Reg] := Low;
368:             While Port[Status_Reg] and Write_Wait <> 0 do;
369:               Port[Data_Reg] := High;
370:           end;
371:         end;
372:       end; (- of Set_Clock_Period -)
373:
374:
375: Procedure Set_Clock_Freq;
376:
377: Var
378:
379:   Period, Actual_Period : extended;
380:
381: Begin
382:
383:   If freq = 0.0 then Actual_Freq := -3.0 (- invalid frequency -)
384:   else
385:     begin

```

```

386:      Period := 1.0 / Freq;
387:      Set_Clock_Period(Period, Actual_Period);
388:      If Actual_Period < 0.0 then Actual_Freq := Actual_Period
389:      else Actual_Freq := 1.0 / Actual_Period;
390:      end;
391:
392: End; (- of Set_Clock_Freq -)
393:
394:
395:
396:
397: Procedure AO_Value;
398:
399: Var
400:
401:     Temp, Low, High : Byte;
402:     Factor, UniVolts, BiVolts, Data_Value : Extended;
403:
404:
405: Begin
406:
407:     Factor := ( 10.0 / Base_Factor );
408:     (- Stop the DT2818 -)
409:     Port[Command_Reg] := CStop;
410:     Temp := Port[Data_Reg];
411:     While Port[Status_Reg] and Write_Wait <> 0 do;
412:     While Port[Status_Reg] and Command_Wait = 0 do;
413:     Port[Command_Reg] := CClear;
414:     While Port[Status_Reg] and Write_Wait <> 0 do;
415:     While Port[Status_Reg] and Command_Wait = 0 do;
416:     Port[Command_Reg] := CADin;
417:     While Port[Status_Reg] and Write_Wait <> 0 do;
418:     Port[Data_Reg] := Gain_Code;
419:     While Port[Status_Reg] and Write_Wait <> 0 do;
420:     Port[Data_Reg] := Channel;
421:     While Port[Status_Reg] and Read_Wait = 0 do;
422:     Low := Port[Data_Reg];
423:     While Port[Status_Reg] and Read_Wait = 0 do;
424:     High := Port[Data_Reg];
425:     Data_Value := Low + 256.0 * High;
426:     While Port[Status_Reg] and Write_Wait <> 0 do;
427:     While Port[Status_Reg] and Command_Wait = 0 do;
428:     Status := Port[Status_Reg];
429:     If Status and $80 = $80 then Voltage := -999.0 (- ie a board error - )
430:     else
431:         Voltage := Data_Value * Factor * 2.0 - 10.0
432: End; (- of AO_Value -)
433:
434: Procedure DA_Value;
435:
436: Var
437:     Temp, Low, High : Byte;
438:     Factor : extended;
439:     DataValue : word;
440:

```

```

441: Begin
442:   Factor := (20.0 / Base_Factor);
443:   {- Stop the DT2818 -}
444:   Port[Command_Reg] := CStop;
445:   Temp := Port[Data_Reg];
446:   While Port[Status_Reg] and Write_Wait <> 0 do;
447:   While Port[Status_Reg] and Command_Wait = 0 do;
448:   Port[Command_Reg] := CClear;
449:   While Port[Status_Reg] and Write_Wait <> 0 do;
450:   While Port[Status_Reg] and Command_Wait = 0 do;
451:   Port[Command_Reg] := CDRout;
452:   While Port[Status_Reg] and Write_Wait <> 0 do;
453:   Port[Data_Reg] := Channel;
454:   DataValue := Round((Voltage + 10) / 20 * 4096);
455:   High := DataValue div 256;
456:   Low := DataValue mod 256;
457:   Voltage := ((High * 256 + Low) / 4096 * 20) - 10;
458:   While Port[Status_Reg] and Write_Wait <> 0 do;
459:   Port[Data_Reg] := Low;
460:   While Port[Status_Reg] and Write_Wait <> 0 do;
461:   Port[Data_Reg] := High;
462:   While Port[Status_Reg] and Write_Wait <> 0 do;
463:   While Port[Status_Reg] and Command_Wait = 0 do;
464:   Status := Port[Status_Reg];
465:   If Status and $80 = $80 then Voltage := -999.0; {- ie a board error - }
466: End; {- of DA_Value -}
467:
468:
469: Procedure Setup_ADC;
470:
471: Var
472:   CountH, CountL : byte;
473:
474: Begin
475:
476:   While Port[Status_Reg] and Command_Wait = 0 do ;
477:   Port[Command_Reg] := CSAD;
478:   While Port[Status_Reg] and Write_Wait <> 0 do ;
479:   Port[Data_Reg] := Gain_Code;
480:   While Port[Status_Reg] and Write_Wait <> 0 do ;
481:   Port[Data_Reg] := 0;
482:   While Port[Status_Reg] and Write_Wait <> 0 do ;
483:   Port[Data_Reg] := End_Channel;
484:   CountH := Count div 256;
485:   CountL := Count mod 256;
486:   While Port[Status_Reg] and Write_Wait <> 0 do ;
487:   Port[Data_Reg] := CountL;
488:   While Port[Status_Reg] and Write_Wait <> 0 do ;
489:   Port[Data_Reg] := CountH;
490:
491: End; {- of Setup_ADC -}
492:
493: Procedure AD_Series;
494:
495: Var

```



```

496:   Error, Loop, ReadL, ReadH : word;
497:
498:
499:   Begin
500:     Setup_ADC(End_Channel, Count);
501:     Get_DT_Error(Error);
502:     If Error <> 0 then exit;
503:     While Port[Status_Reg] and Command_Wait = 0 do;
504:       Port[Command_Reg] := CRAD;
505:       For loop := 0 to Count - 1 do
506:         Begin
507:           While Port[Status_Reg] and Read_Wait = 0 do;
508:             ReadL := Port[Data_Reg];
509:             While Port[Status_Reg] and Read_Wait = 0 do;
510:               ReadH := Port[Data_Reg];
511:               Data_Array[loop] := ReadL + 256 * ReadH;
512:             End;
513:             While Port[Status_Reg] and Command_Wait = 0 do ;
514:               Status := Port[Status_Reg];
515:             End;
516:
517: Procedure Begin_AD_DMA;
518:
519:   Var
520:     Offset, Error, Count : word;
521:     CTemp, Page : byte;
522:
523:   Begin
524:     DMA_Error := 0;
525:     If NewParams then begin
526:       Set_Clock_Freq(Freq, Actual_Freq);
527:       If Not(Ext_Clock) then If Actual_Freq < 0 then DMA_Error := DMA_Error + 1;
528:       If (End_Channel < 0) or (End_Channel > 3) then DMA_Error := DMA_Error + 2;
529:     end;
530:     Count := Scans * (End_Channel + 1);
531:     If Count > (DTArraySize + 1) then DMA_Error := DMA_Error + 4;
532:     If DMA_Error <> 0 then exit;
533:     If NewParams then Setup_ADC(End_Channel, Count);
534:     Get_DT_Error(Error);
535:     If Status and $80 <> 0 then begin DMA_Error := 8 ; exit ; end;
536:     Get_Page_and_Offset(ArrayPtr, Page, Offset);
537:     Port[DMA_Mode] := Write_1;
538:     Port[DMA_Clr_BP] := 0;
539:     Port[DMA_Addr_1] := Offset mod 256;
540:     Port[DMA_Addr_1] := Offset div 256;
541:     Port[DMA_Byte_1] := ((Count * 2) - 1) mod 256;
542:     Port[DMA_Byte_1] := ((Count * 2) - 1) div 256;
543:     Port[DMA_Page_1] := Page;
544:     Port[DMA_Mask] := DMA_Channel;
545:     While Port[Status_Reg] and Command_Wait = 0 do ;
546:       CTemp := CDMA + CRAD;
547:       If Ext_Clock then CTemp := CTemp + CExt_Clock;
548:       If Ext_Irig then CTemp := CTemp + CExt_Irig;
549:       Port[Command_Reg] := CTemp;
550:       Get_DT_Error(Error);

```



```

551:   If Status and $80 (> 0 then begin DMA_Error := 16; exit ; end;
552: End; {- of Begin_AD_DMA -}
553:
554: Function DMA_Count;
555:
556:   Var Temp : word;
557:
558:   Begin
559:     Port[DMA_Clr_BP] := 0;
560:     If Channel = 1 then begin
561:       Temp := Port[DMA_Byte_1];
562:       DMA_Count := Temp + 256 * Port[DMA_Byte_1];
563:     end
564:     else
565:       begin
566:         Temp := Port[DMA_Byte_3];
567:         DMA_Count := Temp + 256 * Port[DMA_Byte_3];
568:       end ;
569:   End; {- of DMA_Count -}
570:
571:
572: Procedure Wait_for_DMA;
573:
574:   Var Hour, Min, Sec, Sec100 : word;
575:       Start_Time           : extended;
576:       Current_Time         : extended;
577:
578:   Begin
579:     GetTime(Hour, Min, Sec, Sec100);
580:     Start_Time := Seconds(Hour, Min, Sec, Sec100);
581:     Status := 0;
582:     Repeat
583:       Begin
584:         GetTime(Hour, Min, Sec, Sec100);
585:         Current_Time := Seconds(Hour, Min, Sec, Sec100);
586:         If Current_Time - Start_Time > DMA_Time_Out then
587:           begin
588:             Status := 1;
589:             exit;
590:           end;
591:       End;
592:     Until DMA_Count(DMA_Channel) = 65535;
593:   End; {- of Wait_for_DMA -}
594:
595: Function Voltage;
596:
597:   Var Temp : extended ;
598:
599:   Begin
600:     Temp := Analog;
601:     Voltage := (Temp / 204.8) - 10;
602:   End; {- of Voltage -}
603:
604: End.
605:

```

## LISTING FOR FILE C:\PASCAL\UNIT\AWPSUNIT.PAS

```

1:  Unit AWPSUnit;
2:
3:  {-----}
4:  {-      Amplitude-Weighted Phase Structuration Developmental Unit      -}
5:  {-                                           -}
6:  {-      Version 22.0      9th August 1989      -}
7:  {-                                           -}
8:  {-                                           -}
9:  {-                                           -}
10: {-      Department of Physical Chemistry      -}
11: {-      University of Bristol      -}
12: {-      Cantock's Close      -}
13: {-      Bristol      -}
14: {-      BS8 1TS      -}
15: {-----}
16:
17: {-----}
18: {- This unit contains various fundamental procedures that are required for -}
19: {- the main Amplitude-Weighted Phase Structuration Package and should reside -}
20: {- in the Turbo Pascal 4.0 unit directory as a compiled .TPU file along side -}
21: {- this source code.      -}
22: {-----}
23:
24: {- This version is for use with MASTER22.PAS -}
25:
26:
27: {$N+}
28:
29: Interface
30:
31: Uses DI2818, Derivs, Crt;
32:
33: Var CentreOffset : extended;
34:
35: Procedure FetchData(var Phase, Amplitude : QuickArray;
36:                    var Histogram, CorrelChan : LongIntArray;
37:                    var Av_Amp : extended;
38:                    NumPoints : word;
39:                    var Time : extended;
40:                    JunkBatch : boolean);
41:
42: Procedure Calculate_AWPS(var Amplitude : QuickArray;
43:                         var OPhase, AWPS, AWPD : LongIntArray;
44:                         var RMSAmp, MeanAmp : longint;
45:                         NumPoints : word;
46:                         Weighting : boolean);
47:
48: Implementation
49:
50: Type LookUpTable = Array[1..254, 1..254] of byte;
51:    LookUpPtr    = ^LookUpTable;
52:
53: Var
54:    Amp_LookUp    : LookUpPtr;
55:    Phase_LookUp  : LookUpPtr;

```

```

56:   InputFile : File of LookUpTable;
57:
58:
59: Procedure FetchData(...);
60:
61:   Var
62:     Data_Array, Data_Array1, Data_Array2 : D1Array;
63:     Data_ArrayPtr : D1ArrayPtr;
64:     DMA_Error : byte;
65:     Freq, Actual_Freq : extended;
66:     NumPointsx6, bin, error, loop, loopx3 : word;
67:     Accepted : boolean;
68:     SinPhi, CosPhi, AmplitudeAv, YMax : extended;
69:     Phi, TempAmp, TempPhi, IntSinPhi, IntCosPhi : integer;
70:     HistoBin : LongInt;
71:     Toggle : boolean;
72:
73:   Begin
74:     Freq := 1 / Time;
75:     Toggle := Not(Toggle);
76:     NumPointsx6 := NumPoints * 6;
77:     Repeat
78:       Accepted := True;
79:       If Toggle then Data_ArrayPtr := @Data_Array1 else
80:         Data_ArrayPtr := @Data_Array2;
81:       If JunkBatch then (- ie first run -)
82:         If (Usable_Length(Data_ArrayPtr) < (Numpoints * 6)) and
83:           (Usable_Length(Data_ArrayPtr) > 0) then
84:           begin
85:             TextMode(3);
86:             Writeln('Insufficient memory for DMA transfer. ',
87:               Usable_Length(Data_ArrayPtr), ' free bytes. ');
88:             Halt;
89:           End;
90:       Begin_AD_DMA(Freq, Actual_Freq, 2, Numpoints, Data_ArrayPtr,
91:         DMA_Error, JunkBatch);
92:       If DMA_Error <> 0 then begin
93:         TextMode(3);
94:         Writeln('*** Begin_AD_DMA error #', DMA_Error);
95:         If DMA_Error = 16 then begin
96:           Get_DI_Error(Error);
97:           Print_Error_Msg(Error);
98:         end;
99:         Halt;
100:       end;
101:       If JunkBatch then If Not(Ext_Clock) then Time := 1 / Actual_Freq;
102:       If Toggle then Move(Data_Array2, Data_Array, NumPointsx6) else
103:         Move(Data_Array1, Data_Array, NumPointsx6);
104:       for loop := 1 to NumPointsx6 div 2 do
105:         Case Data_Array[loop-1] of
106:           0,4095 : Accepted := False;
107:         end;
108:       If Not(Accepted) and Not(JunkBatch) then
109:         begin
110:           Wait_for_DMA;

```

```

111:      If Status = 1 then begin
112:          TextMode(3);
113:          WriteLn('*** DMA Time-out: ',DMA_Count(1));
114:          Halt;
115:      end;
116:      TextColor(Red);
117:      GotoXY(70,25);
118:      Write('OVERLOAD');
119:      Sound(200);
120:      Delay(100);
121:      NoSound;
122:      TextColor(Black);
123:      GotoXY(70,25);
124:      Write('      '*Chr(8));
125:      TextColor(Cyan);
126:      Toggle := Not(Toggle);
127:      end;
128:  Until (Accepted or JunkBatch);
129:  for loop := 0 to Numpoints-1 do
130:      begin
131:          loopx3 := loop*2 + loop;
132:          IntSinPhi := Data_Array[loopx3] div 16;  {- PSD 1 o/p -}
133:          IntCosPhi := Data_Array[loopx3+1] div 16; {- PSD 2 o/p -}
134:          (- CorrelChan[loop] := 2048 - Data_Array[loopx3+2];-)
135:          Phase[loop] := Phase_LookUp^[IntSinPhi, IntCosPhi];
136:          Amplitude[loop] := Amp_LookUp^[IntSinPhi, IntCosPhi];
137:          CorrelChan[loop] := (Amplitude[loop] * Amplitude[loop]) div 256;
138:          ($IFDEF DQHISTOGRAM)
139:              If loop > 0 then Begin
140:                  HistoBin := (Phase[loop] - Phase[loop - 1]);
141:                  HistoBin := (HistoBin * Amplitude[loop] div 2) + 75;
142:                  If (HistoBin > 0) and (HistoBin < 151) then Inc(Histogram[HistoBin]);
143:              End;
144:          ($ELSE)
145:              Inc(Histogram[Amplitude[loop] div 2]);
146:          ($ENDIF)
147:      end;
148:  End; (- FetchData -)
149:
150:
151:  Procedure Calculate_AWPS(...);
152:
153:  { ----- }
154:  { This procedure calculates }
155:  { }
156:  {  $AWPD[\tau] := \delta Q(\tau) = \int_0^{\tau} A(t) \dot{\phi}(t) dt$  }
157:  { }
158:  { }
159:  { and }
160:  { }
161:  { }
162:  {  $AWPS[\tau] := \delta Q(\tau)^2 = \int_0^{\tau} A(t) \dot{\phi}(t)^2 dt$  }
163:  { }
164:  { }
165:  { }

```



```

166:  { using the Trapezoidal method for approximating definite integrals.      }
167:  { ----- }
168:
169:  Var
170:    tau, t : word;
171:    integral, temp1, temp2 : longint;
172:    DTau : extended;
173:
174:  Begin
175:    for tau := 0 to NumPoints - 3 do begin
176:      temp1 := Amplitude[tau];
177:      MeanAmp := MeanAmp + temp1;
178:      RMSAmp := RMSAmp + temp1 * temp1;
179:    end;
180:    integral := 0;
181:    for tau := 1 to NumPoints - 3 do begin
182:      {-
183:        temp1 := Amplitude[tau];
184:        temp2 := DPhase[tau];
185:        integral := integral + temp1 * temp2;
186:      -}
187:      integral := integral + Amplitude[tau] * DPhase[tau];
188:      AUPD[tau] := AUPD[tau] + integral;
189:      AUPS[tau] := AUPS[tau] + (integral div 256) * (integral div 256);
190:    end;
191:  End; {- of Calculate_AUPS -}
192:
193:  Begin {- Initialisation section -}
194:    DMA_Time_Out := 5;
195:    Ext_Clock := false;
196:    Ext_Irig := true;
197:    CentreOffset := 0;
198:    DA_Value(0, CentreOffset);
199:    ResetDT;
200:    ClearDT;
201:    Writeln('');
202:    Writeln('Preconfiguring DT2818 board: DMA Timeout = ', DMA_Time_Out:1:2, 's');
203:    Writeln('          Clock = Internal');
204:    Writeln('          Trigger = External');
205:    Writeln('');
206:    Writeln('Reading amplitude look-up table. Please wait. ');
207:    New(Amp_LookUp);
208:    New(Phase_LookUp);
209:    {$I+}
210:    Assign(InputFile, 'C:\AUPS\LU_TBL\AMP_LU.TBL');
211:    Reset(InputFile);
212:    Read(InputFile, Amp_LookUp^);
213:    Close(InputFile);
214:    Writeln('Reading phase look-up table. Please wait. ');
215:    Assign(InputFile, 'C:\AUPS\LU_TBL\PHASE_LU.TBL');
216:    Reset(InputFile);
217:    Read(InputFile, Phase_LookUp^);
218:    Close(InputFile);
219:    {$I-}
220:  End. {- of AUPSUnit -.}

```



## LISTING FOR FILE C:\PASCAL\UNIT\LEASTSQ4.PAS

```

1:  unit LeastSqr;
2:
3:  {-----}
4:  {-                                             -}
5:  {-   Turbo Pascal Numerical Methods Toolbox   -}
6:  {-   Copyright (c) 1986, 87 by Borland International, Inc. -}
7:  {-                                             -}
8:  {- This unit provides procedures for modelling data with a function of a -}
9:  {- given type, given a set of data points. -}
10: {-                                             -}
11: {-----}
12:
13: ($N+)
14:
15: interface
16:
17: type
18:   Float = extended;
19:
20: const
21:   TNNearlyZero = 1E-015;
22:
23:   TNRowSize = 5;           { Maximum number of terms }
24:                           { in Least Squares fit   }
25:   TNColumnSize = 150;      { Maximum number of data points }
26:
27: type
28:   TNColumnVector = array[1..TNColumnSize] of Float;
29:   TNRowVector = array[1..TNRowSize] of Float;
30:   TNmatrix = array[1..TNColumnSize] of TNRowVector;
31:   TNSquareMatrix = array[1..TNRowSize] of TNRowVector;
32:   TNString40 = string[40];
33:   FitType = byte;
34:
35: function ModuleName(Fit : FitType) : TNString40;
36:
37: procedure LeastSquares(NumPoints      : integer;
38:                        var XData       : TNColumnVector;
39:                        var YData       : TNColumnVector;
40:                        we, lag         : extended;
41:                        NumTerms        : integer;
42:                        var Solution     : TNRowVector;
43:                        var YFit        : TNColumnVector;
44:                        var Residuals    : TNColumnVector;
45:                        var StandardDeviation : Float;
46:                        var Variance     : Float;
47:                        var Error       : byte;
48:                        Fit             : FitType);
49:
50: {-----}
51: {-                                             -}
52: {-   Input: NumPoints, XData, YData, NumTerms   -}
53: {-   Output: Solution, YFit, Residuals, StandardDeviation, Error -}
54: {-                                             -}
55: {- Purpose: Given NumPoints data points of the form (X, Y), this -}

```

```

56:  {-      procedure finds the least square solution of NumTerms terms -}
57:  {-      (NumTerms (= NumPoints) to then matrix equation AC = B -}
58:  {-      where A is a NumPoints by NumTerms matrix, B is a -}
59:  {-      NumPoints vector and C is the least squares solution. The -}
60:  {-      elements of A are A[i, j] = Ij(X[i]) where Ij is the jth -}
61:  {-      basis vector and X[i] is the X-value of the ith data point. -}
62:  {-      The basis vectors are created by a separate include file, -}
63:  {-      or module. The choice of module will determine whether -}
64:  {-      the least squares solution is a polynomial fit, trigono- -}
65:  {-      metric fit (Fourier series), power fit (e.g. Y=ax^b, where -}
66:  {-      b is fractional), exponential fit (e.g. Y=a-Exp(bx)), or -}
67:  {-      logarithmic fit (e.g. Y=a-Ln(bx)). The user may also -}
68:  {-      create modules for other functional forms. See the -}
69:  {-      documentation for details. -}
70:  {-      -}
71:  {- User-Defined Types: INColumnVector = array[1..INColumnSize] of real; -}
72:  {-      INRowVector = array[1..INRowSize] of real; -}
73:  {-      -}
74:  {-      Global Variables: NumPoints      : integer; Number of data points -}
75:  {-      XData          : INColumnVector; X-value data -}
76:  {-      YData          : INColumnVector; Y-value data -}
77:  {-      NumTerms       : integer; Number of terms in -}
78:  {-      least squares fit -}
79:  {-      Solution       : INRowVector; Least squares -}
80:  {-      solution in the -}
81:  {-      given basis -}
82:  {-      YFit           : INColumnVector; Y-values -}
83:  {-      predicted by -}
84:  {-      the LS solution -}
85:  {-      Residuals      : INColumnVector Difference -}
86:  {-      between predicted -}
87:  {-      and actual Y values -}
88:  {-      StandardDeviation : real; Root of variance -}
89:  {-      Error           : byte; Indicates an error -}
90:  {-      -}
91:  {-      Errors: 0: No errors -}
92:  {-      1: NumPoints < 2 -}
93:  {-      2: NumTerms < 1 -}
94:  {-      3: NumTerms > NumPoints -}
95:  {-      4: solution not possible (exact reason -}
96:  {-      will depend upon the particular basis) -}
97:  {-      -}
98:  {-      -----}
99:
100:  implementation
101:
102:  ($I Least1_4.inc)    ( Include procedure code )
103:
104:  ($I Least2.inc)
105:
106:  end. ( LeastSqr )

```

## LISTING FOR FILE C:\PASCAL\INC\LEAST1\_4.INC

```

1:  {-----}
2:  {-                                             -}
3:  {-   Turbo Pascal Numerical Methods Toolbox   -}
4:  {-   Copyright (c) 1986, 87 by Borland International, Inc. -}
5:  {-                                             -}
6:  {-----}
7:
8:  function ModuleName(Fit : FitType) : TNSString40;
9:  begin
10:     ModuleName := ' RUPS fit ';
11:  end; { function ModuleName }
12:
13:
14:  procedure UserTransform(NumPoints      : integer;
15:                          var XData      : TNColumnVector;
16:                          var YData      : TNColumnVector;
17:                          var DummyMultiplier : Float;
18:                          var DummyConstant  : Float;
19:                          var WData      : TNColumnVector;
20:                          var ZData      : TNColumnVector;
21:                          var Error      : byte);
22:
23:  {-----}
24:  {- Input : NumPoints, XData, YData, DummyMultiplier, -}
25:  {-       DummyConstant                               -}
26:  {- Output: WData, ZData, Error                       -}
27:  {-                                             -}
28:  {- This procedure transforms the input data to an appropriate -}
29:  {- format. The transformed (or possibly unchanged) data is -}
30:  {- returned in WData, ZData.                               -}
31:  {-----}
32:
33:  var
34:     Index : integer;
35:
36:  begin
37:     WData := XData;    { No transformation }
38:     ZData := YData;    { No transformation }
39:  end; { procedure UserTransform }
40:
41:  procedure UserInverseTransform(DummyMultiplier : Float;
42:                                DummyConstant    : Float;
43:                                var YFit         : Float);
44:
45:  {-----}
46:  {- Input: DummyMultiplier, DummyConstant, YFit -}
47:  {- Output: YFit                               -}
48:  {-                                             -}
49:  {- This procedure undoes the transformation of -}
50:  {- the YFit values. No inverse transformation -}
51:  {- may be necessary.                           -}
52:  {-----}
53:
54:  begin
55:  end; { procedure UserInverseTransform }

```



```

111:           DummyConstant : float;
112:       var NewSolution    : TNRovVector);
113:
114:       {-----}
115:       {- Input: NumTerms, OldSolution, DummyMultiplier,      -}
116:       {-      DummyConstant                                     -}
117:       {- Output: NewSolution                                     -}
118:       {-                                                         -}
119:       {- This procedure transforms the solution into an appropriate -}
120:       {- form. The transformed (or possibly unchanged) solution  -}
121:       {- is returned in NewSolution.                             -}
122:       {-----}
123:
124:       begin
125:         NewSolution := OldSolution; { No transformation }
126:       end; { procedure UserTransformSolution }

```



## LISTING FOR FILE C:\PASCAL\INC\LEAST2.INC

```

1:  {-----}
2:  {-                                           -}
3:  {-   Turbo Pascal Numerical Methods Toolbox   -}
4:  {-   Copyright (c) 1986, 87 by Borland International, Inc. -}
5:  {-                                           -}
6:  {-----}
7:
8:  procedure LeastSquares(NumPoints      : integer;
9:                        var XData       : TNColumnVector;
10:                       var YData       : TNColumnVector;
11:                       we, lag         : extended;
12:                       NumTerms        : integer;
13:                       var Solution     : TNRowVector;
14:                       var YFit        : TNColumnVector;
15:                       var Residuals    : TNColumnVector;
16:                       var StandardDeviation : float;
17:                       var Variance     : float;
18:                       var Error        : byte;
19:                       Fit              : fitType);
20:
21:  var
22:    WData : TNColumnVector;    { Transformed X-values  }
23:    ZData : TNColumnVector;    { Transformed Y-values  }
24:    Basis : TNmatrix;          { Matrix of basis functions }
25:
26:    Multiplier : float;        { Multiplier and Constant are used in }
27:    Constant : float;          { some modules to pass information  }
28:                                { from Transform to InverseTransform }
29:                                { or TransformSolution. These must   }
30:                                { therefore have dummy parameters    }
31:                                { when Multiplier and Constant are    }
32:                                { not used.                            }
33:
34:  procedure InitializeAndFormBasisVectors(NumPoints : integer;
35:                                          var XData   : TNColumnVector;
36:                                          var YData   : TNColumnVector;
37:                                          var Multiplier : float;
38:                                          var Constant  : float;
39:                                          var WData    : TNColumnVector;
40:                                          var ZData    : TNColumnVector;
41:                                          var NumTerms  : integer;
42:                                          var Solution  : TNRowVector;
43:                                          var Basis     : TNmatrix;
44:                                          var Error     : byte;
45:                                          we, lag      : extended;
46:                                          Fit          : fitType);
47:
48:  {-----}
49:  {- Input: NumPoints, XData, NumTerms          -}
50:  {- Output: Solution, Error                    -}
51:  {-                                           -}
52:  {- This procedure initializes Solution and Error -}
53:  {- to zero. It also checks the data for errors. -}
54:  {-----}
55:  begin

```

```

56:   Error := 0;
57:   if NumPoints < 2 then
58:     Error := 1; { Less than 2 data points }
59:   if NumTerms < 1 then
60:     Error := 2; { Less than 1 coefficient in the fit }
61:   if NumTerms > NumPoints then
62:     Error := 3; { Number of data points less than
63:                 { number of terms in Least Squares fit! }
64:   FillChar(Solution, SizeOf(Solution), 0);
65:   if Error = 0 then
66:   begin
67:     UserTransform(NumPoints, XData, YData, Multiplier,
68:                  Constant, WData, ZData, Error);
69:     if Error = 0 then
70:       UserCreateBasisFunctions(NumPoints, NumTerms, WData, Basis, we, lag, Fit);
71:   end;
72: end; { procedure InitializeAndFormBasisVectors }
73:
74: procedure CreateAndSolveEquations(NumPoints : integer;
75:                                   NumTerms : integer;
76:                                   var Basis : TNmatrix;
77:                                   var ZData : TNColumnVector;
78:                                   var Solution : TNRowVector;
79:                                   var Error : byte);
80:
81:
82: var
83:   Coefficients : TNSquareMatrix;
84:   Constants : TNRowVector;
85:
86:
87: procedure ComputeNormalEquations(NumPoints, NumTerms : integer;
88:                                   var Basis : TNmatrix;
89:                                   var YData : TNColumnVector;
90:                                   var Coefficients : TNSquareMatrix;
91:                                   var Constants : TNRowVector);
92:
93:
94: var
95:   Row, Column, Index : integer;
96:   Sum : Float;
97:
98: begin
99:   for Column := 1 to NumTerms do
100:   begin
101:     Sum := 0;
102:     for Index := 1 to NumPoints do
103:       Sum := Sum + YData[Index] * Basis[Index, Column];
104:       Constants[Column] := Sum;
105:       for Row := Column to NumTerms do
106:       begin
107:         Sum := 0;
108:         for Index := 1 to NumPoints do
109:           Sum := Sum + Basis[Index, Row] * Basis[Index, Column];
110:           Coefficients[Row, Column] := Sum;

```

```

111:     Coefficients[Column, Row] := Sum;
112:     end;
113:     end;
114: end; { procedure ComputeNormalEquations }
115:
116: {-----}
117:
118: procedure Partial_Pivoting(Dimen      : integer;
119:                            Coefficients : TNSquareMatrix;
120:                            Constants    : TNRowVector;
121:                            var Solution  : TNRowVector;
122:                            var Error     : byte);
123:
124:
125:     procedure Initial(Dimen      : integer;
126:                      var Coefficients : TNSquareMatrix;
127:                      var Constants    : TNRowVector;
128:                      var Solution     : TNRowVector;
129:                      var Error        : byte);
130:
131:
132:     begin
133:         Error := 0;
134:         if Dimen < 1 then
135:             Error := 1
136:         else
137:             if Dimen = 1 then
138:                 if ABS(Coefficients[1, 1]) < TNNearlyZero then
139:                     Error := 2
140:                 else
141:                     Solution[1] := Constants[1] / Coefficients[1, 1];
142:                 end; { procedure Initial }
143:
144:                 procedure EROswitch(var Row1 : TNRowVector;
145:                                     var Row2 : TNRowVector);
146:
147:
148:                 var
149:                     DummyRow : TNRowVector;
150:
151:                 begin
152:                     DummyRow := Row1;
153:                     Row1 := Row2;
154:                     Row2 := DummyRow;
155:                 end; { procedure EROswitch }
156:
157:                 procedure EROmultAdd(Multiplier : Float;
158:                                     Dimen      : integer;
159:                                     var ReferenceRow : TNRowVector;
160:                                     var ChangingRow  : TNRowVector);
161:
162:
163:                 var
164:                     Tern : integer;
165:

```

```

166: begin
167:   for Term := 1 to Dimen do
168:     ChangingRow[Term] := ChangingRow[Term] + Multiplier * ReferenceRow[Term];
169:   end; { procedure EROmult&Add }
170:
171:   procedure UpperTriangular(Dimen      : integer;
172:                             var Coefficients : TNSquareMatrix;
173:                             var Constants   : TNRVector;
174:                             var Error      : byte);
175:
176:
177:   var
178:     Multiplier : Float;
179:     Row, ReferenceRow : integer;
180:
181:     procedure Pivot(Dimen      : integer;
182:                    ReferenceRow : integer;
183:                    var Coefficients : TNSquareMatrix;
184:                    var Constants   : TNRVector;
185:                    var Error      : byte);
186:
187:
188:   var
189:     PivotRow, Row : integer;
190:     Dummy : Float;
191:
192:   begin
193:     { First, find the row with the largest element }
194:     PivotRow := ReferenceRow;
195:     for Row := ReferenceRow + 1 to Dimen do
196:       if ABS(Coefficients[Row, ReferenceRow]) >
197:         ABS(Coefficients[PivotRow, ReferenceRow]) then
198:         PivotRow := Row;
199:     if PivotRow <> ReferenceRow then
200:       { Second, switch these two rows }
201:       begin
202:         EROswitch(Coefficients[PivotRow], Coefficients[ReferenceRow]);
203:         Dummy := Constants[PivotRow];
204:         Constants[PivotRow] := Constants[ReferenceRow];
205:         Constants[ReferenceRow] := Dummy;
206:       end
207:     else
208:       { If the diagonal element is zero, no solution exists }
209:       if ABS(Coefficients[ReferenceRow, ReferenceRow]) < TNNearlyZero then
210:         Error := 2; { No solution }
211:     end; { procedure Pivot }
212:
213:   begin { procedure UpperTriangular }
214:     { Make Coefficients matrix upper triangular }
215:     ReferenceRow := 0;
216:     while (Error = 0) and (ReferenceRow < Dimen - 1) do
217:       begin
218:         ReferenceRow := Succ(ReferenceRow);
219:         { Find row with largest element in this column }
220:         { and switch this row with the ReferenceRow }

```



```

221:   Pivot(Dimen, ReferenceRow, Coefficients, Constants, Error);
222:
223:   if Error = 0 then
224:     for Row := ReferenceRow + 1 to Dimen do
225:       { Make the ReferenceRow element of these rows zero }
226:       if ABS(Coefficients[Row, ReferenceRow]) > INHearlyZero then
227:         begin
228:           Multiplier := -Coefficients[Row, ReferenceRow] /
229:             Coefficients[ReferenceRow, ReferenceRow];
230:           EROmultAdd(Multiplier, Dimen,
231:             Coefficients[ReferenceRow], Coefficients[Row]);
232:           Constants[Row] := Constants[Row] +
233:             Multiplier * Constants[ReferenceRow];
234:         end;
235:       end; { while }
236:       if ABS(Coefficients[Dimen, Dimen]) < INHearlyZero then
237:         Error := 2; { No solution }
238:       end; { procedure UpperTriangular }
239:
240:   procedure BackwardsSub(Dimen      : integer;
241:     var Coefficients : TNSquareMatrix;
242:     var Constants    : TNRovUector;
243:     var Solution     : TNRovUector);
244:
245:   var
246:     Tern, Row : integer;
247:     Sum : Float;
248:
249:   begin
250:     Tern := Dimen;
251:     while Tern >= 1 do
252:       begin
253:         Sum := 0;
254:         for Row := Tern + 1 to Dimen do
255:           Sum := Sum + Coefficients[Tern, Row] * Solution[Row];
256:         Solution[Tern] := (Constants[Tern] - Sum) / Coefficients[Tern, Tern];
257:         Tern := Pred(Tern);
258:       end;
259:     end; { procedure BackwardsSub }
260:
261:   begin { procedure Partial_Pivoting }
262:     Initial(Dimen, Coefficients, Constants, Solution, Error);
263:     if Dimen > 1 then
264:       begin
265:         UpperTriangular(Dimen, Coefficients, Constants, Error);
266:         if Error = 0 then
267:           BackwardsSub(Dimen, Coefficients, Constants, Solution);
268:         end;
269:       end; { procedure Partial_Pivoting }
270:
271:   {-----}
272:
273:   begin { procedure CreateAndSolveEquations }
274:
275:

```



```

276:   { The following procedure computes Coefficients and      }
277:   { Constants of the normal equations. The exact          }
278:   { solution to the square system of normal equations     }
279:   { will be the least squares fit to the data.            }
280:   ComputeNormalEquations(NumPoints, NumTerms, Basis, ZData,
281:       Coefficients, Constants);
282:   Partial_Pivoting(NumTerms, Coefficients, Constants, Solution, Error);
283:   if Error = 2 then { Returned from Partial_Pivoting }
284:       Error := 4; { No solution }
285: end; { procedure CreateAndSolveEquations }
286:
287: procedure TransformSolutionAndFindResiduals(NumPoints      : integer;
288:       NumTerms      : integer;
289:       var YData      : INColumnVector;
290:       var Solution    : INRowVector;
291:       Multiplier     : Float;
292:       Constant       : Float;
293:       var Basis       : INmatrix;
294:       var YFit        : INColumnVector;
295:       var Residuals   : INColumnVector;
296:       var StandardDeviation : Float;
297:       var Variance    : float);
298:
299:
300: procedure ComputeYFitAndResiduals(NumPoints      : integer;
301:       NumTerms      : integer;
302:       Multiplier     : Float;
303:       Constant       : Float;
304:       var YData      : INColumnVector;
305:       var Solution    : INRowVector;
306:       var Basis       : INmatrix;
307:       var YFit        : INColumnVector;
308:       var Residuals   : INColumnVector;
309:       var StandardDeviation : float;
310:       var Variance    : float);
311:
312:
313: var
314:   Index, Term : integer;
315:   Sum : float;
316:
317: begin
318:   Sum := 0;
319:   for Index := 1 to NumPoints do
320:   begin
321:     YFit[Index] := 0;
322:     for Term := 1 to NumTerms do
323:       YFit[Index] := YFit[Index] + Solution[Term]*Basis[Index, Term];
324:
325:       { The next procedure undoes the transformation of      }
326:       { the YFit values. For example, if ZData=Ln(YData)     }
327:       { then InverseTransform performs the function          }
328:       { YFit[Index] := Exp(YFit[Index]) so that YFit may     }
329:       { be compared to YData.                                }
330:       UserInverseTransform(Multiplier, Constant, YFit[Index]);

```

```

331:     Residuals[Index] := YFit[Index] - YData[Index];
332:     Sum := Sum + Sqr(Residuals[Index]);
333:   end;
334:   Variance := Sum;
335:   if NumPoints = NumTerms then
336:     StandardDeviation := 0
337:   else
338:     StandardDeviation := Sqrt(Sum/(NumPoints - NumTerms));
339:   end; { procedure ComputeYFitAndResiduals }
340:
341:   begin { procedure TransformSolutionAndFindResiduals }
342:     ComputeYFitAndResiduals(NumPoints, NumTerms, Multiplier, Constant,
343:       YData, Solution, Basis, YFit,
344:       Residuals, StandardDeviation, Variance);
345:     UserTransformSolution(NumTerms, Solution, Multiplier, Constant,
346:       Solution);
347:   end; { procedure TransformSolutionAndFindResiduals }
348:
349:   begin { procedure LeastSquares }
350:     InitializeAndFormBasisVectors(NumPoints, XData, YData,
351:       Multiplier, Constant, WData, ZData,
352:       NumTerms, Solution, Basis, Error, we, lag, Fit);
353:     if Error = 0 then
354:       CreateAndSolveEquations(NumPoints, NumTerms, Basis, ZData,
355:         Solution, Error);
356:
357:     if Error = 0 then
358:       TransformSolutionAndFindResiduals(NumPoints, NumTerms, YData, Solution,
359:         Multiplier, Constant, Basis, YFit,
360:         Residuals, StandardDeviation, Variance);
361:
362:   end; { procedure LeastSquares }

```

## LISTING FOR FILE C:\PASCAL\UNIT\DERIVS.PAS

```

1:  Unit Derivs;
2:
3:  {$N+}
4:
5:  Interface
6:
7:  Type
8:    DerivArray = array [0..150] of extended;
9:    QuickArray = array [0..150] of byte;
10:   LongIntArray = array [0..150] of longint;
11:
12:  Const
13:    Undefined : extended = 9.999999999E+35;
14:
15:  Procedure Derivatives(var XData, YData, DYData, D2YData : DerivArray;
16:                        Numpoints : word);
17:
18:  Procedure First_Derivative(var XData, YData, DYData : DerivArray;
19:                             Numpoints : word);
20:
21:  Function Tan(x: extended) : extended;
22:
23:  Implementation
24:
25:  Function Tan;
26:
27:  Begin
28:    Tan := sin(x)/cos(x);
29:  End; {- of Tan -}
30:
31:  Procedure Derivatives{(var XData, YData, DYData, D2YData : DerivArray;
32:                        NumPoints : word)};
33:  Var
34:    x : integer;
35:    Alpha_plus_two_beta, alpha, alpha_plus_beta : extended;
36:    Delta_x : extended;
37:
38:  Begin
39:    Delta_x := XData[1] - XData[0];
40:    for x := 1 to NumPoints - 2 do
41:    begin
42:      Alpha_plus_two_beta := ArcTan( ( YData[x] - YData[x-1] ) / Delta_x );
43:      Alpha               := ArcTan( ( YData[x+1] - YData[x] ) / Delta_x );
44:      Alpha_plus_beta     := (Alpha + Alpha_plus_two_beta) / 2;
45:      DYData[x] := Tan(Alpha_plus_beta);
46:    end;
47:    DYData[0] := Undefined;
48:    DYData[NumPoints - 1] := Undefined;
49:    for x := 2 to Numpoints - 3 do
50:    begin
51:      Alpha_plus_two_beta := ArcTan( ( DYData[x] - DYData[x-1] ) / Delta_x );
52:      Alpha               := ArcTan( ( DYData[x+1] - DYData[x] ) / Delta_x );
53:      Alpha_plus_beta     := (Alpha + Alpha_plus_two_beta) / 2;
54:      D2YData[x] := Tan(Alpha_plus_beta);
55:    end;

```

```

56:   D2YData[0] := Undefined;
57:   D2YData[1] := Undefined;
58:   D2YData[Numpoints - 1] := Undefined;
59:   D2YData[Numpoints - 2] := Undefined;
60: End; (- of Derivatives -)
61:
62: Procedure First_Derivative((var XData, YData, DYData : DerivArray;
63:                             Numpoints : word));
64:
65:   Var
66:     x : integer;
67:     Alpha_plus_two_beta, alpha, alpha_plus_beta : extended;
68:     Delta_x : extended;
69:
70:   Begin
71:     Delta_x := XData[1] - XData[0];
72:     for x := 1 to NumPoints - 2 do
73:       begin
74:         Alpha_plus_two_beta := ArcTan(YData[x] - YData[x-1]);
75:         Alpha                := ArcTan(YData[x+1] - YData[x]);
76:         Alpha_plus_beta      := (Alpha + Alpha_plus_two_beta) / 2;
77:         DYData[x] := Tan(Alpha_plus_beta);
78:       end;
79:       D2YData[0] := Undefined;
80:       D2YData[NumPoints - 1] := Undefined;
81:     End; (- of First_Derivative -)
82:
83:   End.
84:
85:
86:
87:

```

## LISTING FOR FILE C:\QBASIC3\5210DATA.BAS

```

1:  ' Program to fetch 5210 setup status for use by AUPS software
2:  ' Parameters are stored in C:\5210DATA.OUT in ASCII format, each parameter
3:  ' occupying a separate line
4:
5:  OPEN "COM1:9600,E,7,1,CS,DS" FOR RANDOM AS #1
6:  OPEN "C:\5210DATA.OUT" FOR OUTPUT AS #3
7:  CR$ = CHR$(13)
8:  RESTORE
9:  WHILE 1
10:     READ B$
11:     IF B$ = "00" THEN CLOSE : END
12:     B$ = B$ + CR$
13:     GOSUB SENDSTRING
14:     GOSUB RECEIVESTRING: PRINT #3, LEFT$(Z$, LEN(Z$) - 1);
15:  WEND
16:
17:  SENDSTRING:
18:  WHILE LOC(1) > 0: A$ = INPUT$(1, #1): WEND
19:  FOR J = 1 TO LEN(B$)
20:     C$ = MID$(B$, J, 1): PRINT #1, C$;
21:  WHILE LOC(1) = 0: WEND
22:  A$ = INPUT$(1, #1)
23:  NEXT J
24:  RETURN
25:
26:  RECEIVESTRING:
27:  A$ = "": Z$ = ""
28:  WHILE (A$ <> "") AND A$ <> "?"
29:     Z$ = Z$ + A$
30:  WHILE LOC(1) = 0: WEND
31:  A$ = INPUT$(1, #1)
32:  WEND
33:  RETURN
34:
35:
36:  FETCHSTATUS:
37:  B$ = "ST" + CR$
38:  GOSUB SENDSTRING
39:  GOSUB RECEIVESTRING
40:  STATUS = VAL(Z$)
41:  RETURN
42:
43:  END
44:
45:
46:  DATA ATC, DR, FF, FLT, FRQ, G, ID, LF, OR, OF, REMOTE, RS, SEN, TC, VER, XDB
47:  DATA 00
48:
49:  ' Output file therefore consists of:-
50:  ' FILTER TUNING MODE (0 OR 1)
51:  ' DYNAMIC RESERVE (0 TO 2)
52:  ' FILTER MODE (0 TO 3)
53:  ' REF. FREQ (IN MHz)
54:  ' VERNIER GAIN (0 OR 1 , GAIN)
55:  ' ID (i.e. 5210)

```



A2-81

56: \* LINE FILTERS (0 TO 3)  
57: \* OSC. AMPLITUDE (IN mV)  
58: \* OSC. FREQUENCY (VALUE 0 TO 20000, BAND 0 TO 5)  
59: \* REMOTE STATUS (0 OR 1)  
60: \* RS232 PARAMETERS (BAUD RATE 0 TO 12, OTHERS 0 TO 31)  
61: \* SENSITIVITY (0 TO 15)  
62: \* TIME CONSTANT (0 TO 13)  
63: \* 5210 SOFTWARE VERSION  
64: \* OUTPUT FILTER ROLLOFF (0 OR 1)  
65:

The PALS software is a user-friendly menu-driven program which allows the user to perform real time PALS experiments, generating simultaneous amplitude histograms, AWPD and AWPS functions and auto-correlation functions for the real (or imaginary) component of the demodulated Doppler signal. The experiments can be performed in either a synchronous or continuous mode (see § 3.6.4).

The main features of the software are the data acquisition and processing, data storage/retrieval and the constant velocity minimisation procedure. The principle of operation of the latter has been described in § 4.2.13.

Data acquisition is achieved using the DMA facilities of the ADC board and the host computer. The required number of data point sets are transferred to an array in the host computer's memory. Two arrays are used, with each batch of data being transferred alternately to each array. Whilst data is being transferred to one array, the data in the previous array is processed. In order to achieve real time processing, all the mathematical operations are performed using integer arithmetic. The necessary amplitude and phase data are not calculated from the raw data, but are obtained using look-up tables. Two 254 by 254 8-bit integer arrays reside in memory and act as look-up tables. The raw data from the lock-in amplifier is scaled to 8-bits, yielding numbers ranging from 0 to 255. 0 and 255 represent the

maximum output levels from the LIA, and, hence, are inherently ambiguous. Any sets of data containing 0 or 255 are rejected by the software. This provides a means of minimising interference by dust in the sample being studied, since scattering from dust usually results in strong Doppler signals, which overload the LIA. The two look-up tables yield the necessary amplitude and phase data, with both being scaled from 0 to 255. The use of look-up tables and integer arithmetic allows the processing time to be less than the acquisition time for sampling times down to 1 ms. Below 1ms, the processing time dominates the total experimental duration, and true real time performance is not obtained. After the required number of sets of data have been acquired (e.g. 500), the various experimental functions are normalised. If a series of experiments are being run consecutively, then, between each run, the AutoTrack facility is employed to minimise the constant velocity contribution to the next run and least squares analysis is performed on the various experimental functions to obtain the desired velocity information. If chosen, all of the experimental data is written as a binary file to disc and a text file, summarising the experiment, is written to disc or sent to a printer. The binary files can be retrieved by the software if reprocessing at a future date is required. The structure of the binary file can be seen in "DEF\_VARS. INC".

If the electrode drive amplitude is non-zero, then synchronous averaging is employed. Continuous averaging is employed for zero electrode drive amplitude.

## REFERENCES

1. D.H. Everett, "Basic Principles of Colloid Science", Royal Society of Chemistry, London, 1988.
2. D.J. Shaw, "Introduction to Colloid and Surface Chemistry", 3rd ed., Butterworth's, London, 1980.
3. B. Vincent, Adv. Coll. Int. Sci., 4, 193 (1974)
4. D.H. Napper, J. Coll. Int. Sci., 58, (2), 390 (1977)
5. B.V. Deryaguin and L. Landau, Acta. Phys. Chim. U.R.S.S., 14, 633 (1941)
6. E.J.W. Verwey and J. Th. G. Overbeek, "Theory of Stability of Lyophobic Colloids", Elsevier, 1948.
7. P.W. Atkins, "Physical Chemistry", 2nd ed., Oxford University Press, Oxford, 1982.
8. H.C. Hamaker, Physica, 4, 1058 (1937)
9. G.Gouy, J. Phys., 9, 457 (1910)
10. D.L. Chapman, Phil. Mag., 25, 475 (1913)
11. O. Stern, Z. Electrochem, 30, 508 (1924)
12. J. Th. Overbeek, "Electrokinetic Phenomena" in "Colloid Science", H.R. Kruyt ed., Vol I, p194. Elsevier, 1949.
13. A.L. Loeb, J.Th. G. Overbeek and P.H. Wiersema, "The Electrical Double Layer Around a Spherical Colloidal Particle", MIT Press, Cambridge, Mass., 1961.
14. R. Hogg, T.W. Healy and D.W. Fuerstenau, Trans. Farad. Soc., 62, 1638 (1966)
15. H. Reerink and J. Th. G Overbeek, Discuss. Faraday Soc., 18, 74 (1954)
16. M. Von Smoluchowski, Z. Phys. Chem., 92, 129 (1917)
17. H.R. Kruyt, "Colloid Science", Vol I, Elsevier, 1952.



18. J. Lyklema, Adv. Coll. Int. Sci., 2, 65 (1968)
19. G.D. Parfitt, J.A.Wood and R.T. Ball, J. Chem. Soc., Faraday I, 69, 1908 (1973)
20. D.N.L. McGown and G.D. Parfitt, Discuss. Faraday, Soc., 42, 225 (1966)
21. B.C. Soyenkoff, J. Phys. Chem., 35(4), 2993 (1931)
22. E.J.W. Verwey, Rev. Trav. Chim., 60, 625 (1941)
23. D.N.L. McGown, G.D. Parfitt and E. Willis, J. Coll. Int. Sci., 20, 650 (1965)
24. L.A. Romo, J. Phys. Chem., 67, 386 (1963)
25. F.J. Micale, Y.K. Lui and A.C. Zettlemoyer, Discuss. Faraday Soc., 42, 238 (1966)
26. V.R. Damerell and A. Urbanic, J. Phys. Chem., 48, 125 (1944)
27. V.R. Damerell and R. Mattson, J. Phys. Chem., 48, 134 (1944)
28. V.R. Damerell, K. Gayer and H. Laudenschlager, J. Phys. Chem., 49, 436 (1945)
29. K.E. Lewis and G.D. Parfitt, Trans. Faraday Soc., 62, 1652 (1966)
30. H.C. Parreira, J. Electroanal. Chem., 25, 69 (1970)
31. R.M. Fuoss, J. Am. Chem. Soc., 80, 5059 (1958)
32. M.E. Labib and R. Williams, J. Coll. Int. Sci., 97(2), 356 (1984)
33. M.E. Labib and R. Williams, Coll. Polym. Sci., 264, 533 (1986)
34. M.E. Labib, Coll. Surfaces, 29, 293 (1988)
35. R. Schmid, J. Solution Chem., 12, 135 (1983)
36. M. Van Der Waarden, J. Coll. Sci., 5, 317 (1950)
37. J.J. Chessick, A.C. Zettlemoyer and G.J. Young, J. Coll. Sci., 13, 372 (1958)
38. G.D. Parfitt and T.D. Wiltshire, J. Phys. Chem., 68, 3545 (1964)
39. F.M. Fowkes, J. Coll. Sci., 20, 664 (1965)
40. G.D. Parfitt, J. Coll. Sci., 20, 664 (1965)

41. O. Griot, Trans. Faraday Soc., 62, 2904 (1966)
42. C.A. Malbrel and P. Somasundaran, J. Coll. Int. Sci., 133(2), 404 (1989)
43. J.W. Goodwin, F. McDonald and P.A. Reynolds, Coll. Surfaces, 33, 1 (1988)
44. J.W. Goodwin, N. Casson and P.A. Reynolds, "Dispersions in Non-Aqueous Media", Conference at Shell Thornton Research Centre, April 1990.
45. J.W. Goodwin, Personal Communication (1990)
46. Ph. C. van der Hoeven and J. Lyklema, "Dispersions in Non-Aqueous Media", Conference at Shell Thornton Research Centre, April 1990.
47. Ph. C. van der Hoeven, Personal Communication (1990)
48. S.S. Dukhin and B.V. Deryaguin, "Electrokinetic Phenomena" in "Surface and Colloid Science", E. Matijevic ed., Vol. 7, Wiley-Interscience, New York, 1974.
49. F.F. Reuss, Memoires de la Society Imperiale des Naturalistes de Moscou, 2, 327 (1809)
50. G. Quincke, Pogg. Ann., 107, 1 (1859)
51. M. Von Smoluchowski in "Handbuch der Electrilität und des Magnetismus", (Graetz), Vol II, 366, Barth, Leipzig, 1921.
52. D.C. Henry, Proc. Roy. Soc. A, 133, 106 (1931)
53. P. Debye and E. Hückel, Physik, Z., 25, 49 (1924)
54. E. Hückel, Physik. Z., 25, 204 (1924)
55. D.C. Henry, Trans. Faraday Soc., 44, 1021 (1948)
56. R.J. Hunter, "Zeta Potential in Colloid Science", Academic Press, London, 1981.
57. J.J. Hermans, Trans. Faraday Soc., 36, 133 (1940)
58. J. Th. G. Overbeek, Adv. Coll. Sci., 3, 97 (1950)
59. P.H. Wiersema, A.L. Loeb and J.Th.G. Overbeek, J. Coll. Int. Sci., 22, 78 (1966)

60. R.W. O'Brien and L.R. White, J. Chem. Soc., Faraday II, 74, 1607 (1978)
61. D. Stigter and K.J. Mysels, J. Phys. Chem., 59, 45 (1955)
62. S. Levine and G.M. Bell, J. Phys. Chem., 64, 1188 (1960)
63. H. Ohshima, T.W. Healy and L.R. White, J. Coll. Int. Sci., 90(1), 17 (1982)
64. H. Ohshima, T.W. Healy and L.R. White, J. Chem. Soc., Faraday Trans. 2, 79, 1613 (1983)
65. M. Bier ed., "Electrophoresis", Vol I and II, Academic Press, New York, 1959 & 1967.
66. A.D. Bangham, R. Flemans, D.H. Heard and G.V.F. Seaman, Nature, 182, 642 (1958)
67. C.J. Van Oss, R.M. Fike, R.J. Good and J.M. Reinig, Analyt. Biochem., 60, 242 (1974)
68. V.J. Novotny, "Reprographic Technology", Chapter 14, A.C.S. Symposium Series, 200, 281 (1982)
69. J.L. Van Der Minne and P.H.J. Hermanie, J. Coll. Int. Sci., 7, 600 (1952)
70. S. Kuo and F. Osterle, J. Coll. Int. Sci., 25, 421 (1967)
71. P.S. Vincett, J. Coll. Int. Sci., 76, 83 (1980)
72. H.C. Parreira, J. Chem. Phys., 49(10), 4711 (1968)
73. T.A. Vorob'eva, I.N. Vlodavets and S.S. Dukhin, Colloid J. USSR, 32, 152 (1970)
74. E. Delatour and M. Hanss, Rev. Sci. Instruments, 47, 1531 (1976)
75. A.K. Gaigalas, S. Woo and J.B. Hubbard, J. Coll. Int. Sci., 136(1), 213 (1990)
76. B.H. Zimm, R.S. Stein and P. Doty, Polymer Bulletin, 1, 90 (1945)
77. E.R. Pike, Phil. Trans. R. Soc. Lond. A, 293, 349 (1979)
78. R. Hanbury-Brown and R.Q. Twiss, Nature, 177, 27 (1956)
79. R. Pecora, J. Chem. Phys., 40(6), 1604 (1964)
80. B. Chu, "Laser Light Scattering", Academic Press, New York, 1974.



81. P.N. Pusey, Phil. Trans. R. Soc. Lond. A, 293, 429 (1979)
82. K. Schätzel, Appl. Phys. B, 42, 193 (1987)
83. L. Mandel, Prog. Opt., 2, 181 (1963)
84. A.J.F. Siegert, MIT Rad. Lab. Report No. 465 (1943)
85. E.S. Swinbourne, J. Chem. Soc., Part II, 2371 (1960)
86. P.N. Pusey, D.E. Koppel, D.W. Schaefer, R.D. Camerini-Otero and R.M. Franklin, J. Phys. (Paris), 33, C1 (1972)
87. G.A. Korn and T.M. Korn, "Mathematical Handbook for Scientists and Engineers", 2nd ed., 643, McGraw-Hill, New York, 1968.
88. Appendix D in ref. 87
89. A.T. Forrester, R.A. Gudmundsten and P.O. Johnson, Phys. Rev., 99, 1691 (1955)
90. B.R. Ware and W.H. Flygare, Chem. Phys. Lett., 12(1), 81 (1971)
91. B.R. Ware, Adv. Coll. Int. Sci., 4, 1 (1974)
92. B.R. Ware and W.H. Flygare, J. Coll. Int. Sci., 39, 670 (1972)
93. Y. Yeh and H.Z. Cummins, Appl. Phys. Lett., 4, 176 (1964)
94. T. Yoshimura, A. Kikkawa and N. Suzuki, Japanese. J. Appl. Phys., 11(12), 1797 (1972)
95. F. Durst, A. Melling and J.H. Whitelaw, "Principles and Practice of Laser-Doppler Anemometry", 2nd ed., Academic press, London. 1981.
96. M.J. Rudd, J. Phys. E, 2, 55 (1969)
97. F. Durst and J.H. Whitelaw, Opto-Electronics, 5, 137 (1973)
98. L.E. Drain, "The Laser-Doppler Technique", J. Wiley, Chichester, 1980.
99. H.H. Bossel, W.J. Hiller and G.E.A. Meier, J. Phys. E., Sci. Instr., 9, 893 (1972)

100. R. Dändliker and P.D. Iten, Appl. Optics., 13, 286 (1974)
101. P. Debye and F.W. Sears, Proc. Nat. Acad. Sci., 18, 409 (1932)
102. K. Schätzel and J. Merz, J. Chem. Phys., 81(5), 2482 (1984)
103. K. Schätzel, Rev. Roum. Phys., 32, 37 (1987)
104. J.F. Miller, K. Schätzel and B. Vincent, accepted for publication in J. Coll. Int. Sci.
105. K. Schätzel, Personal communication
106. D.A. Parkins, Ph.D. Thesis, Bath (1986)
107. A.W. Preece and N.P. Luckman, Phys. Med. Biol., 26(1), 11 (1981)
108. E.E. Uzgiris, Prog. Surface Sci., 10, 53 (1981)
109. Isomet specification sheet for Model 1205C acousto-optic modulator
110. Supplied by courtesy of K.Ryan, Bristol University
111. See Hewlett Packard 3582A Spectrum Analyser Operating Manual, pp 3-69 to 3-72 (Manual Part No. 03582-90005)
112. "9412A Phase-Sensitive Detector Handbook", Publication No. M9412A1, Brookdeal Electronics Ltd., UK, 1974.
113. L.F. Phillips, "Electronics for Experimenters in Chemistry, Physics and Biology", Wiley, New York, 1966.
114. "DT2818 User Manual", Documents UM-00666-E-2991 and UM-03112-A, Data Translation Inc., 1985
115. "Turbo Pascal 4.0 Owner's Handbook", Borland International, California. 1987.
116. "Model 5210 Lock-in Amplifier Instruction Manual", Part No. 4200-0325, 19874-A-MNL-A, Princeton Applied Research Corp., 1988.
117. Supplied for measurement by A.J. Milling, Bristol University
118. G.J. Janz and R.P.T. Thomas, "Non-Aqueous Electrolytes Handbook", Vols I & II, Academic Press New York, 1972.
119. CRC Handbook of Chemistry and Physics, 67th Ed., CRC Press, Florida, 1986.
120. Metrohm 652KF-Coulometer Operating Manual, Metrohm Limited., Switzerland, 1982.



121. A.I. Medalia, J. Polymer Sci., 6, 423 (1951)
122. J. Clarke, Ph.D.Thesis, Bristol University (1980)
123. S. Harley, M.Sc Thesis, Bristol University (1984)
124. W.H. Guthrie, Ph.D Thesis, Lehigh University (1985)
125. William Freeman & Co., UK, Personal communication
126. J.E. Lind Jr. and R.M. Fuoss, J. Phys. Chem., 65, 999 (1961)
127. I.P. MacDonald, Exxon Chemical Ltd., Private communication (1990)
128. J.F. Marsh, Chem. Ind., 470 (1987)

